Winter 2005

Contributions to modeling and computer efficient estimation for Gaussian space -time processes

Veronica Pocsik Hupper
University of New Hampshire, Durham

CONTRIBUTIONS TO MODELING AND COMPUTER EFFICIENT ESTIMATION FOR GAUSSIAN SPACE-TIME PROCESSES

BY

VERONICA POCSIK HUPPER

Bachelor of Arts Degree, Caldwell College, 1996
Master of Science Degree, University of New Hampshire, 2001

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Mathematics

December 2005

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

ABSTRACT

CONTRIBUTIONS TO MODELING AND COMPUTER EFFICIENT ESTIMATION FOR GAUSSIAN SPACE-TIME PROCESSES

by

Veronica Pocsik Hupper

University of New Hampshire, December 2005

This thesis research provides several contributions to computer efficient methodology for estimation with space-time data. First we propose a parsimonious class of computer-efficient Gaussian spatial interaction models that includes as special cases CAR and SAR - like models. This extended class is capable of modeling smooth spatial random fields. We show that, for rectangular lattices, this class is equivalent to higher-order Markov random fields. Thus we capture the computational advantage of iterative updating of Markov random fields, while at the same time provide the possibility of simple interpretation of smooth spatial structure.

This class of spatial models is defined via a spatial structure removing orthogonal transformation, which we propose for any spatial interaction model as a means to improve computation time. Such a transformation is a one-time preprocessing step in iterative estimation, such as in MCMC. For very large data on a rectangular lattice we can achieve further computational savings by circulant embedding which enables use of FFT for calculations. We examine how the model as well as the embedding can be incorporated in hierarchical models for space time data with spatially varying temporal trend components. We describe an application in arctic hydrology where gridded runoff fields are investigated for local trends.

INTRODUCTION

In the past several years, the need for methods to analyze data collected over both space and time has increased, especially in the environmental sciences. The continuing development of satellite imagery, the increasing use of geographic information systems (GIS), and the creation of new technologies in automated data collection produce ever-larger data sets. These data sets include observations taken at each time increment at a set of spatial locations. The analysis of temporal trends and spatial patterns can be considered separately, but in most cases the trends over time are similar at locations that are a short distance from each other, so called "neighbors". Similarities at neighboring locations are usually more than mere coincidence implying that a model exists to explain the relationships. Such similarities at neighboring locations can be statistically modeled resulting in a combined space-time analysis.

Techniques commonly used in the geosciences to address both spatial and temporal trends consist of analyzing the data over time at each individual location. This analysis often involves a set of separate linear regressions relating trends of variables, or simply determining trends of individual variables. Regression parameters such as temporal trends are then plotted over space and analyzed visually. It is also possible to plot the results of significance tests, such as p-values or decisions such as "+", "-", or "0" denoting significant positive, negative, or non-significant parameter estimates, respectively.

There are several drawbacks to such techniques. Determining if there are trends over space is subjective because they are analyzed only visually. More importantly, there is low statistical power in determining the significance of temporal trends estimated in this fashion because they are based on a small number of data points (corresponding to the number of time points over which the observations were recorded).

Another approach to modeling spatiotemporal data is to combine time series models with spatial models such as variogram-based models. Handcock and Wallis (1994) attempted to model mean temperatures over both space and time by fitting first a spatial model. The spatial parameters were then tested for temporal correlation. Similarly, Stroud, Müller, and Sansó (2001) describe an approach using a state space modeling framework where the mean at each time is estimated using a locally weighted linear regression. In this case, the weight function allows for the inclusion of prior information on the unknown spatial process. Neither of these methods allow for simultaneous estimation of both spatial and temporal structure. It should also be noted that any significance tests on the spatial parameters are limited by the number of time points and tests on the temporal parameters by the number of spatial locations, resulting in fewer degrees of freedom.

A combined space-time analysis utilizes all the data to estimate temporal trends at individual locations in addition to estimating parameters of spatial and temporal autoregressive patterns. Wikle, et al (1998) describe a hierarchical model specifically for analyzing environmental data in both space and time. This hierarchical model is a cascade of stages, the first of which describes measurement error with the remaining levels addressing parameters that influence the underlying process. Pettitt, Weir, and

Hart (2002) suggest the use of a Gaussian conditional autoregressive model to explain the spatial cohesion in the underlying spatial process. In this approach, the hierarchical model is described where the underlying trend is assumed to be a Markov random field. Although their setup contains no covariates, their addition does not complicate the model and is an easy extension.

The estimation of parameters in this hierarchical setup is most conveniently done within the Bayesian paradigm using Markov chain Monte Carlo procedures such as the Gibbs sampler, Metropolis-Hastings algorithm, or a combination thereof (Gelman, et al, 2000). Since these methods of estimation are iterative and computer intensive, the size of the data sets highly influences the computation time (and in some cases the ability to estimate the parameters at all). Often, in environmental applications, the data sets can be extremely large preventing successful application of MCMC procedures. In such cases, subdividing the data in space can help solve the computational challenge (Rue, 2001).

The interest here is to modify existing procedures to allow for fast, computationally efficient estimation of parameters in both the spatial and space-time framework and also to provide a better model to represent extremely smooth spatial processes. First, background information is provided on both spatial and temporal methods individually. In particular, Chapter I focuses on Gaussian spatial processes as well as both conditional autoregressive (CAR) models and simultaneous autoregressive (SAR) models. The intrinsic conditional autoregressive model (Besag and Kooperberg, 1995 and Besag, Mollie, and York, 1991) is also considered here. The Pettit, Weir, and Hart (2002) parameterization of the CAR model is also explored here because of its computational efficiency properties. Chapter II focuses on parameter estimation for the Pettitt, Weir and

Hart parameterization of the CAR model. This relies on Markov Chain Monte Carlo methods and so Bayesian techniques such as the Metropolis Hastings algorithm and Gibbs sampling are examined.

Of utmost importance for applications to large data sets is computational efficiency. Among the topics covered in Chapter III is an orthogonal data transformation that removes correlation structure. This process in addition to using circulant embedding will ease much of the computational burden in using MCMC methods to estimate the parameters in the hierarchical model, particularly matrix inversion and eigenvalue and eigenvector computations. In Chapter IV, the CAR model is expanded to include a smoothness parameter so as to be able to better describe smooth spatial processes. Relationship of this extended autoregressive (EAR) model to ordinary CAR models with higher order neighbor structure will be determined.

First-order autoregressive models of time series data are described in Chapter V. A hierarchical model that incorporates properties of both spatial autoregressive models as well as autoregressive models for time series is developed in Chapter VI to model spatially varying temporal trends simultaneously. The EAR model is then incorporated into the space-time setup. Finally, in Chapter VI, the hierarchical space-time model is applied to the river runoff data from the hydrological data bank R-ArcticNET.

CHAPTER I

SPATIAL DATA

Overview

Most spatial data or geospatial data consist of measurements recorded at locations over some spatial region in either the two-dimensional ($\mathbf{R}^2$) or three-dimensional ($\mathbf{R}^3$) Euclidean space. This type of data is treated differently from the usual forms of observational data in that it is of interest to determine if measurements at nearby locations affect the behavior of the data at a given location. Many "classical" analyses such as regression and ANOVA rely on the assumption that observations are independently and identically distributed (iid), an assumption that is violated when spatial dependence exist. These procedures need to be adjusted when applied to spatially collected data, and thus involve the calculation of a spatial interaction or spatial correlation representing the strength of this dependence between nearby locations.

There are three sub-categories of spatial data. When the location itself is of interest, the resulting data can be analyzed for spatial patterns, so-called point patterns. Since the analysis of this kind of spatial data is very different from when observations are collected at fixed spatial locations, spatial point pattern analysis will not be considered any further in this thesis.

The second category arises when data are in the form of spatial "lattice" data. When data is collected in this manner, it is assumed that the data is not continuous over space. Instead, each measurement represents an average or total over a fixed spatial area or

region. An example of such data would be any measurement taken on a town basis. The population of one town, for example, is different from the population in a neighboring town with no gradient of population in between. A sudden change in population value occurs when crossing over town boundaries. Data collected over a lattice is analyzed using spatial interaction models. In this approach, one assumes that neighboring locations are more alike than non-neighbors. These models require knowledge about the lattice configuration, especially the definition of which regions are considered neighbors of a given region.

In the third category, data is continuous over space and is usually modeled using geostatistical methods. This approach models the correlation between any two responses as a function of the distance between the two locations. Since data that is continuous over space is collected at discrete locations, modifications can be made to spatial interaction models to model continuous space data. Such modifications usually involve higher order neighbor structures for regularly gridded data and/or the implementation of a distance based weight structure.

Gaussian Spatial Process

When dealing with a spatial process, observed over continuous or discrete space, it is often convenient to model the data as Gaussian (normally distributed) with a trend in the mean and with spatially correlated error. Thus, spatial distributions are regression models with errors that are not independent and identically distributed.

Usually, in a regression model, the mean is simply $\mathbf{X\beta}$, where $\mathbf{X}$ is the $n \times p$ matrix containing the independent or explanatory variables in the model and the $p \times 1$ vector $\boldsymbol{\beta}$ contains the slope parameters from the regression. The variance is assumed to be

$$\text{var}\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} = \boldsymbol{\Sigma} = \sigma^2 \mathbf{I}.$$

In the case of a process with spatial structure, the mean remains the same, but the variance-covariance matrix must incorporate the spatial dependencies. So, for the spatial model, the variance-covariance matrix is

$$\boldsymbol{\Sigma}_{sp} = \sigma^2 \mathbf{K}(\boldsymbol{\theta})$$

where σ^2 is a constant variance term, $\boldsymbol{\theta}$ is a vector of spatial parameters to be determined by the type of model used to analyze the data, and

$$\mathbf{K}(\boldsymbol{\theta}) = \left\{ corr\left(z_i, z_j\right) \right\} \qquad i, j = 1 \ldots n.$$

Assume that the process is Gaussian. Then

$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \sim N\left(\boldsymbol{\mu} = \mathbf{X\beta}, \boldsymbol{\Sigma} = \sigma^2 \mathbf{K}(\boldsymbol{\theta})\right).$$

The goal of most spatial analyses is to estimate the parameters $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\theta}$. The likelihood of these parameters given the normal distribution above, can be expressed as

$$L\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\right) = \frac{1}{\left(2\pi\right)^{n/2} \left|\boldsymbol{\Sigma}\right|^{1/2}} e^{-\frac{1}{2\sigma^2}(\mathbf{z}-\mathbf{X\beta})' \mathbf{K}(\boldsymbol{\theta})^{-1} (\mathbf{z}-\mathbf{X\beta})},$$

where $|\boldsymbol{\Sigma}| = (\sigma^2)^n |K(\boldsymbol{\theta})|$ denotes the determinant of $\boldsymbol{\Sigma}$. Maximum likelihood estimation typically utilizes the log-likelihood

$$\log L(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log|\mathbf{K}(\boldsymbol{\theta})| - \frac{1}{2\sigma^2}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})' \mathbf{K}(\boldsymbol{\theta})^{-1}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})$$
$$= c - n\log(\sigma) - \frac{1}{2}\log|\mathbf{K}(\boldsymbol{\theta})| - \frac{1}{2\sigma^2}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})' \mathbf{K}(\boldsymbol{\theta})^{-1}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})$$

or

$$-2\log L(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = c + n\log(\sigma) + \log|\mathbf{K}(\boldsymbol{\theta})| + \frac{1}{\sigma^2}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})' \mathbf{K}(\boldsymbol{\theta})^{-1}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})$$

where c is a constant. This model is difficult to use for large data sets (large n) because it is computationally intensive to find both the determinant and the inverse of $\mathbf{K}(\boldsymbol{\theta})$, in repeated evaluations when using numerical maximization to determine the maximum likelihood estimates (Mardia & Marshall, 1984; Mardia & Watkins, 1989). This represents a major drawback for geostatistical methods where distance based models are assumed for the spatial correlation $K(\boldsymbol{\theta})$, such as the "exponential model" where $(K(\boldsymbol{\theta}))_{ij} = e^{-\theta d_{ij}}$ and $d_{ij} = |s_i - s_j|$ is the distance between locations s_i and s_j.

Conditional Autoregressive Model (CAR)

One of the most popular spatial interaction models is the conditional autoregressive or CAR model (Besag, 1974). Here, the data at one location is modeled conditional on the data collected at neighboring locations. So, if we have observations y_i taken at location s_i for $i = 1, 2, ..., n$, one can express the conditional distribution of y_i as

$$(y_i | \mathbf{y}_{-i}) \approx (y_i | \mathbf{y}_j, s_j \in N(i))$$

where $\mathbf{y}_{-i}$ is the collection of all observations that are not y_i and $N(i)$ is the set of neighbors of location s_i.

The condition that y_i given all others $\mathbf{y}_{-i}$ only depends on the neighbors on location s_i, provides a Markov structure in space, and hence these models are called Markov random fields (Besag, 1974). The fact that in the spatial context, y_i given all others in effect implies that y_i only depends on the values observed at locations neighboring s_i indicates that the Markov random field depends on a given neighbor structure, which in essence is a graph that connects all spatial units. A lattice, either regular or irregular, defines such a neighbor structure. However, Markov random fields are very general and allow for so-called "higher order" structure. For example, in a square lattice, the north-south and east-west neighbors are referred to as first-order neighbors while the four nearest diagonal locations are called second-order neighbors. Higher order neighbor structures can be accommodated easily in Markov random fields. Sometimes, a weight function is chosen so that it decreases with higher order, thus losing the effect of measurements taken farther away.

As an alternative, a distance-based weight function can be assumed, that is applied to a central location within each lattice element. A distance-based weight function allows one to apply lattice models to data that is continuous in space, or more typically irregular point data. The uniform, linear, and reciprocal weight functions are suggested by Pettitt, Weir, and Hart (2002). These are based upon a given fixed maximum distance, *rmax*, and d_{ij} is defined to be the distance between locations s_i and s_j.

Uniform	$g(d_{ij}) = \begin{cases} 1 & \text{for } 0 < d_{ij} < r\max \\ 0 & \textit{otherwise} \end{cases}$
Linear	$g(d_{ij}) = \begin{cases} 1 - \dfrac{d_{ij}}{r\max} & \text{for } 0 < d_{ij} < r\max \\ 0 & \textit{otherwise} \end{cases}$
Reciprocal	$g(d_{ij}) = \begin{cases} \dfrac{r\max}{d_{ij}} - 1 & \text{for } 0 < d_{ij} < r\max \\ 0 & \textit{otherwise} \end{cases}$

Assuming that the conditional distribution of the data is normal, the expected value and variance are, respectively,

$$E(y_i|\mathbf{y}_{-i}) = \sum_{i \neq j} c_{ij} y_j$$

and

$$\text{var}(y_i|\mathbf{y}_{-i}) = \sigma_i^2 .$$

Here, $\mathbf{y}_{-i}$ is defined as before. For the joint distribution of all the y's, we require a symmetric positive definite variance matrix, which results in

$$\mathbf{y} \sim N(\mathbf{0}, \mathbf{V})$$

where $\mathbf{V} = \mathbf{Q}^{-1}$, $\mathbf{Q} = \mathbf{M}^{-1}(\mathbf{I} - \mathbf{C})$ and $\mathbf{C} = \{c_{ij}\}$ (Besag, 1974). In this case, each c_{ij} is an interaction term. Various parameterizations for the c_{ij} have been suggested. Notice that the CAR structure provides a direct model for the precision matrix, **Q**, and thus there is no need for the matrix inversion of the variance-covariance matrix, **V**, in the likelihood calculations.

In the simplest form of this model, $\mathbf{M} = \mathbf{\Sigma} = diag(\sigma_i^2, i = 1,...,n)$ is a diagonal matrix with the $\sigma_i^2 = \frac{\sigma^2}{n_i}$ on the diagonal and zeros everywhere else (see also Cliff & Ord, 1981), where n_i is the number of neighbors of location s_i. The matrix $\mathbf{C}$ will incorporate both a spatial interaction term, φ, and the specified weights. For example, the first order conditional autoregressive model can be set up so that $\mathbf{Q} = diag\left(\frac{1}{\sigma_i^2}\right)(\mathbf{I} - \varphi\mathbf{G})$, where σ_i^2 is defined as above. In this case, $\mathbf{C} = \varphi\mathbf{G}$, where $\mathbf{G}$ is the weight matrix defined by one of the weight functions suggested by Pettitt, Weir, and Hart (ie: linear, uniform, or reciprocal).

The case where the joint distribution is not centered at zero is an easy extension. Suppose that

$$\mathbf{y}^* = \mathbf{y} + \boldsymbol{\mu}.$$

Then, the distribution is merely shifted with no change in variability resulting in

$$E\left[y_i \middle| \mathbf{y}_{-i}\right] = \mu_i + \sum_{i \neq j} c_{ij}\left(y_i - \mu_i\right) \tag{1.1}$$

which implies that

$$\mathbf{y}^* \sim MVN(\boldsymbol{\mu}, \mathbf{V}).$$

Here, $\mathbf{V} = \mathbf{Q}^{-1}$ with $\mathbf{Q}$ defined as above. Notice that regardless of where $\mathbf{y}$ is centered, the conditional autoregressive process models spatial relationships in the residuals, not in the variable itself.

Simultaneous Autoregressive Model (SAR)

For Gaussian spatial processes, there exists a simultaneous specification for the mean and variance resulting in the simultaneous autoregressive model (SAR), also referred to

as Whittle's model. Unlike in the CAR model, the data is not modeled conditional on the observations at neighboring locations. Instead, the joint density function of the random variables is expressed as a product of the joint density function of each of the variables with its neighbors.

In a SAR model, the data at one location is assumed to be the weighted average of observations at neighboring points plus random variation. So, the observation at each location can be expressed as

$$y_i = \sum_{i \neq j} g_{ij} y_j + \varepsilon_i, \qquad i = 1, \ldots, n$$

where g_{ij} are "weights" to be specified and the ε_i are uncorrelated error terms with $E(\varepsilon_i) = 0$ and $Var(\varepsilon_i) = \sigma_i^2$. In matrix form, $\mathbf{y} = \mathbf{G}\mathbf{y} + \boldsymbol{\varepsilon}$ where $\mathbf{y}$ is the $n \times 1$ vector containing the data, $\boldsymbol{\varepsilon}$ is the $n \times 1$ error vector, and $\mathbf{G} = \{g_{ij}\}$. Solving for $\mathbf{y}$,

$$\mathbf{y} = (\mathbf{I} - \mathbf{G})^{-1} \boldsymbol{\varepsilon}.$$

results in

$$E(\mathbf{y}) = \mathbf{0}$$

and

$$Var(\mathbf{y}) = (\mathbf{I} - \mathbf{G})^{-1} \boldsymbol{\Sigma} \left[(\mathbf{I} - \mathbf{G})^{-1} \right]^T$$

where $\boldsymbol{\Sigma} = diag(\sigma_i^2)$. Since $\mathbf{G}$ is symmetric and $\mathbf{I}$ is the identity, this can be written as

$$Var(\mathbf{y}) = (\mathbf{I} - \mathbf{G})^{-1} \boldsymbol{\Sigma} (\mathbf{I} - \mathbf{G}^T)^{-1}.$$

Like the conditional autoregressive model, the precision matrix, $\mathbf{Q} = \mathbf{V}^{-1}$, can be expressed directly. In this case,

$$\mathbf{Q} = (\mathbf{I} - \mathbf{G})\, diag\left(\frac{1}{\sigma^2}\right)(\mathbf{I} - \mathbf{G}^T).$$

Assuming that the errors are normally distributed,

$$\mathbf{y} \sim MVN(\mathbf{0}, \mathbf{V}).$$

The case where the mean is nonzero results in

$$\mathbf{y} \sim MVN(\boldsymbol{\mu}, \mathbf{V})$$

where **V** is defined as above. From here, the log likelihood can be found, but computations result in the same difficulties as the previously described likelihood method when the data set is particularly large.

In the case of the SAR model, spatial parameters are incorporated into the matrix **G**. For example, in the case of a first order SAR model, one could write the model as

$$\mathbf{y}_i = \psi \sum_{i \neq j} w_{ij} \mathbf{y}_j + \boldsymbol{\varepsilon}_i .$$

Here, $\mathbf{G} = \psi \mathbf{W}$ where

$$w_{ij} = \begin{cases} 1 & s_j \in N(i) \\ 0 & otherwise \end{cases}$$

Recall that $N(i)$ is the collection of first order neighbors of location s_i.

The Pettitt et. al. Parameterization of the CAR Model

Both the conditional autoregressive and simultaneous autoregressive models are computationally more efficient than geostatistical models since there is no need for repeated inversion of the variance-covariance matrix when likelihood methods are performed that consist of many iterations. However, the issue of finding the determinant of this matrix remains. There are several methods that can be used to attempt this, but

most are cumbersome for large data sets. Computer efficient determinant calculations have only been available for relatively simple parameterizations. Recall that the more spatial locations that are involved, the larger all these matrices become. This is where problems arise when performing matrix operations such as finding determinants, even when using computer software. This issue is addressed in a paper by Pettitt, Weir, and Hart (2002), by Ord (1980), and many other papers.

The model described by Pettitt, Weir, and Hart provides a particular parameterization of the ordinary CAR model described above that proves to be computationally efficient. In this model, the precision matrix is created in such a way that the determinant is computed easily and in closed form. It also lends itself to the addition of covariates without complicating the model much. This computational efficiency is particularly advantageous for large irregular lattices and weighting schemes applied to continuous space data.

Recall that the CAR model results in the data having the joint distribution

$$\mathbf{y} \sim MVN(\mathbf{0}, \mathbf{V})$$

where $\mathbf{V}^{-1} = \mathbf{Q} = \mathbf{M}^{-1}(\mathbf{I} - \mathbf{C})$, $\mathbf{M} = \mathbf{\Sigma} = diag(\sigma_i^2, i = 1,...,n)$, and $\mathbf{C} = \{c_{ij}\}$. The extension to the case where the mean is nonzero is an easy application as shown with the ordinary CAR model. In Pettitt, Weir, and Hart, the matrices **M** and **C** are defined so that the terms of the matrix **C** are

$$c_{ij} = \begin{cases} \dfrac{\phi g_{ij}}{1 + |\phi| \sum_{k \in N(i)} g_{ik}}, & i \neq j \\ 0, & i = j \end{cases}$$

Notice that since **C** incorporates the weight structure, the diagonal will contain all zeros. In the ordinary CAR model, **M** is a diagonal matrix containing the variances. Instead, Pettitt, Weir, and Hart define the terms of **M** as

$$m_{ii} = \frac{1}{1+|\phi| \sum_{k \in N(i)} g_{ik}}, \qquad i = 1, 2, ..., n .$$

Here, g_{ij} denotes the lattice weights, usually of one of the forms described for the ordinary CAR model, and ϕ is a specific interaction parameter. This results in the precision matrix

$$\mathbf{Q} = \mathbf{M}^{-1}(\mathbf{I} - \mathbf{C}) \tag{1.2}$$

with

$$Q_{ij} = \begin{cases} 1+|\phi| \sum_{k \in N(i)} g_{ik} & i = j \\ -\phi g_{ij} & i \neq j \end{cases}.$$

The diagonal dominance criteria states that it is a sufficient but not a necessary condition that a matrix **A** is positive definite if it has the property

$$A_{ii} - \sum_{j:i \neq j} |A_{ij}| > 0 . \tag{1.3}$$

In this case,

$$Q_{ii} = 1+|\phi| \sum_{k \in N(i)} g_{ik}$$

and, since all the value of g_{ij} are positive by definition

$$\sum_{j:i \neq j} |Q_{ij}| = \sum_{j:i \neq j} |-\phi g_{ij}| = |\phi| \sum_{j:i \neq j} g_{ij} .$$

When locations s_i and s_j are neighbors, $g_{ij} > 0$, and when they are not $g_{ij} = 0$. So, the only values for g_{ij} that count in this sum are those for which the two locations are neighbors. Thus

$$Q_{ii} - \sum_{j:i \neq j} |Q_{ij}| = 1 + |\phi| \sum_{k \in N(i)} g_{ik} - |\phi| \sum_{k \in N(i)} g_{ik} = 1.$$

It is obvious that **Q** is symmetric because of the symmetry in the neighbor relationship. Therefore, the precision matrix is symmetric and positive-definite for all values $-\infty < \phi < \infty$, making it a valid precision matrix.

Conditions for positive definiteness of **Q** in the ordinary CAR parameterization (Ord, 1975) often lead to restrictions on the parameter space that depend on the lattice structure. Thus, in this form suggested by Pettitt, et. al., **Q** can be rewritten

$$\mathbf{Q} = \mathbf{I} - \phi(\mathbf{G} - \mathbf{D}) \tag{1.4}$$

where

$$\mathbf{G} = \left(g\left(d_{ij}\right)\right) \tag{1.5}$$

and

$$\mathbf{D} = diag\left(\sum_{s_k \in N(i)} g\left(d_{ik}\right)\right). \tag{1.6}$$

The eigenvalues of **Q** can then be found such that

$$EV(\mathbf{Q}) = \mathbf{I} - \phi EV(\mathbf{G} - \mathbf{D}) \tag{1.7}$$

where $EV(\mathbf{X})$ represents the eigenvalues of the matrix **X**. Once the eigenvalues are found, the determinant of **Q** can be easily obtained as the product of its eigenvalues.

A singular value decomposition (SVD) of $\mathbf{G}-\mathbf{D}$ will result in the desired eigenvalues and eigenvectors. It is assumed that the factorization

$$\mathbf{F}\Lambda\mathbf{F}^{\mathrm{T}} = \mathbf{G}-\mathbf{D}$$

exists with $\Lambda = diag(\lambda_i)$, where λ_i is the i^{th} eigenvalue of $\mathbf{G}-\mathbf{D}$. The columns of $\mathbf{F}$ contain the corresponding eigenvectors. Most mathematics or statistics software packages such as Matlab or S-PLUS have built in functions to factor matrices into its singular value decomposition. Once the eigenvalues, λ_i, of $\mathbf{G}-\mathbf{D}$ are found, the eigenvalues of $\mathbf{Q}$ can be expressed as

$$\kappa_i = 1-\phi\lambda_i .$$

and the determinant of $\mathbf{Q}$ can be easily computed as

$$|\mathbf{Q}| = \prod_{i=1}^{n}(1-\phi\lambda_i).$$

Intrinsic Conditional Autoregressive Model

The intrinsic CAR, or "pair wise difference" CAR, as it is often called, is used extensively in Bayesian hierarchical models for disease mapping (Besag, Mollie, and York, 1991). This model is also implemented in the software WINBUGS, a program which does Bayesian analysis using the Gibbs Sampler.

Recall that in the Gaussian CAR model it is assumed that the spatial process has the joint distribution

$$\mathbf{z} \sim MVN(\boldsymbol{\mu}, \mathbf{V})$$

where μ is some mean vector and $\mathbf{V} = \mathbf{Q}^{-1}$ is the variance covariance matrix. Thus the density can be written as

$$p(\mathbf{z}) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}} e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \mathbf{Q}(\mathbf{z}-\boldsymbol{\mu})}.$$

Here the precision matrix, $\mathbf{Q}$, is a positive definite matrix. Using the identity (Besag and Kooperberg, 1995)

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} = \sum_{\forall i} Q_{i+} x_i^2 - \sum_{i<j} Q_{ij}\left(x_i - x_j\right)^2$$

the density becomes

$$p(\mathbf{z}) = \frac{1}{(2\pi)^{n/2}|\mathbf{V}|^{1/2}} e^{-\frac{1}{2}\left[\sum_{\forall i} Q_{i+}(z_i-\mu_i)^2 - \sum_{i<j} Q_{ij}\left[(z_i-z_j)-(\mu_i-\mu_j)\right]^2\right]}$$

where Q_{i+} represents the i^{th} row sum.

The intrinsic CAR is the limiting form of the Gaussian CAR where $\mathbf{Q}$ is well defined but $\mathbf{Q1} = \mathbf{0}$ (Besag and Kooperberg, 1995). This implies that the row sums of $\mathbf{Q}$ are all zero and that the joint density of the spatial process $\mathbf{z}$ is now

$$p(\mathbf{z}) \propto e^{-\frac{1}{2}\sum_{i<j} Q_{ij}\left[(z_i-z_j)-(\mu_i-\mu_j)\right]^2}.$$

Under this assumption, $\mathbf{Q}$ is no longer positive definite but instead is positive semi-definite and $\mathbf{V} = \mathbf{Q}^{-1}$ no longer exists.

When applied to the computer efficient CAR model described by Pettitt, Weir, and Hart (2002),

$$\lim_{\phi\to\infty} |\phi|^{-1} Q_{ij} = \begin{cases} \sum_{k\in N(i)} g_{ik} & i = j \\ -sign(\phi) g_{ij} & i \neq j \end{cases}$$

results in

$$E\left(z_i \left| \mathbf{z}_{-i} \right.\right) = \mu_i + \frac{1}{\sum_{j \in N(i)} g_{ij}} \sum_{j \in N(i)} g_{ij}\left(z_j - \mu_j\right)$$

and

$$Var\left(z_i \left| \mathbf{z}_{-i} \right.\right) = \frac{\sigma^2}{\sum_{j \in N(i)} g_{ij}}.$$

Although the joint mean and variance do not exist for **z**, the conditional mean and variance are quite easy to work with. Thus this model is useful as a prior specification for a residual spatial process where there is a constraint that all residuals sum to zero.

CHAPTER II

PARAMETER ESTIMATION FOR SPATIAL PROCESSES

Overview

Recall that in general, the Gaussian spatial model assumes the joint distribution

$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \sim N\left(\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma} = \sigma^2 \mathbf{K}(\boldsymbol{\theta})\right)$$

resulting in the log likelihood

$$-2\log L\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\right) = n\log\left(2\pi\right) + n\log\left(\sigma^2\right) + \log\left|\mathbf{K}(\boldsymbol{\theta})\right| + \frac{1}{\sigma^2}\left(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\right)' \mathbf{K}(\boldsymbol{\theta})^{-1}\left(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\right)$$

Assuming a one parameter spatial interaction model with interaction parameter ϕ, the substitution

$$\mathbf{K}(\boldsymbol{\theta}) = \mathbf{K}(\phi) = \mathbf{V} = \mathbf{Q}^{-1}$$

can be made. Thus, the log likelihood becomes

$$-2\log L\left(\boldsymbol{\beta}, \sigma^2, \phi\right) = n\log\left(2\pi\right) + n\log\left(\sigma^2\right) - \log\left|\mathbf{Q}\right| + \frac{1}{\sigma^2}\left(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\right)^T \mathbf{Q}\left(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\right).$$

Notice that this requires taking the determinant of the matrix $\mathbf{Q}$. Recall that $\mathbf{Q}$ is an $n \times n$ square matrix where n is the number of spatial locations. For traditional CAR and SAR models, finding the determinant of $\mathbf{Q}$ can be computationally taxing for irregular lattices. Pettitt, Weir, and Hart (2002) define $\mathbf{Q}$ so that its determinant can be written

$$\left|\mathbf{Q}\right| = 1 - \phi\left|\mathbf{G} - \mathbf{D}\right|$$

where $\mathbf{G}$ is the weight matrix and is sparse when dealing with lattices and only low order neighbors. So, given $|\mathbf{G}|$, it is easy to compute $|\mathbf{Q}|$.

Mardia and Marshall (1984), Kitanidis and Lane (1985), and Mardia and Watkins (1989) all looked at using maximum likelihood methods to estimate the parameters of a spatial model. In all cases, maximum likelihood estimation was used in conjunction with numerical methods to estimate the covariance structure of linear models over a spatial region. Here, it was noticed that the maximum likelihood approach can be successful for estimating covariance parameters for a small to moderate number of spatial locations only. For much larger contemporary spatial data sets like those arising from remote sensing and computer-based data collection techniques, maximum likelihood methods would not be the best way to estimate spatial parameters. Instead, it is suggested that Monte Carlo methods are better suited for estimating spatial parameters over a large spatial region. In fact one can use the conditional specification of the CAR model to generate iteratively univariate conditional distributions. With many such iterations, the resulting simulated values represent draws from the joint distribution of the y's. This is in effect a Gibbs sampler. It was in the early developments of Markov random fields (Besag, 1974, Geman and Geman, 1984) where Gibbs sampling originated in order to solve the then computationally difficult problem of drawing from the joint distribution of a spatial random field.

Markov Chain Monte Carlo (MCMC) Estimation

Estimating parameters in complex statistical models often results in computationally difficult calculations. Models such as spatial hierarchical models have many parameters. For such models, finding sampling distributions of parameters using a frequentist approach is analytically intractable. So, a Bayesian approach is used to describe the spatial parameters. But, even using Bayesian methods, posterior distributions are often analytically intractable. Finding marginal distributions for a single parameter can require difficult integrations over the parameter space. For a model with many parameters and applied to data collected over large spatial regions, any analytic method of parameter estimation can be difficult. In most instances, no exact solutions exist.

An alternative to obtaining sampling distributions analytically is to obtain posterior distributions for each parameter by way of simulations, so-called Monte-Carlo methods. One approach is to use direct sampling. This involves taking random draws from a distribution and comparing it to values from the joint posterior. One such method is the rejection method (Rice, 1995 or Chib and Greenberg, 1995). This method is commonly used to generate random variables from a density function, but can be extended to the generation of parameter values from a joint posterior distribution easily. The parallels are obvious. Let $p(\boldsymbol{\theta}|\mathbf{y})$ be the joint posterior distribution of k parameters given the data, $\mathbf{y}$. It is assumed that the joint posterior, p, is nonzero on some subset of $\mathbf{R}^k$, $\mathbf{S}$, and is zero outside this region. Note that p can be nonzero on all of $\mathbf{R}^k$ as in the multivariate normal in k dimensions. Let M be a function in k dimensions that is always greater than p in $\mathbf{S}$, and is zero outside of $\mathbf{S}$. Then the function m, defined as

$$m(\mathbf{x}) = \frac{M(\mathbf{x})}{\int_S M(\mathbf{x})d\mathbf{x}},$$

is also a probability function. The idea is to choose M so that it is easy to generate values from m. If $\mathbf{S}$ is not all of $\mathbf{R}^k$ and does not extend infinitely in any of the k dimensions, then M can be chosen in such a way that m is the k dimensional uniform distribution over $\mathbf{S}$. The idea is to generate $\mathbf{w}$ with the distribution m and u from the uniform distribution on $[0,1]$, independent of $\mathbf{w}$. If $M(\mathbf{w}) \cdot \mathrm{u} \le p(\mathbf{w})$, then $\mathbf{w}$ is accepted. If $\mathbf{w}$ is not accepted, a new $\mathbf{w}$ and a new u need to be generated to repeat the process until an acceptable $\mathbf{w}$ is found. In order for this method to be computationally efficient, there needs to be a relatively high probability of acceptance. If the probability is too low, a large number of iterations will need to be performed in order to find a single acceptable vector $\mathbf{w}$. When k is large, this may prove difficult and is impossible without knowing the joint posterior distribution.

An alternate approach is to use Markov chain sampling. This approach is often successful for simulating from posterior distributions arising from hierarchical models (Gelman et al, 1997). It requires the creation of a Markov chain that approaches a stationary distribution, the desired posterior distribution. The basic idea behind Markov chain simulation is to draw possible values for $\boldsymbol{\theta}$ from a jumping or transition distribution, which defines the transition probability of the Markov chain. Here, $\boldsymbol{\theta}$ is a vector containing all the parameters in the model. For each iteration, generated values of $\boldsymbol{\theta}$ are accepted in a similar manner as in the rejection method. What makes this different from direct sampling is that the distributions of successive values of $\boldsymbol{\theta}$ are dependent on the previous vector drawn. Thus, the sequence of values forms a Markov chain. The key

here is that after a suitable number of iterations, referred to as the "burn-in period", the Markov chain has reached its steady state, and the relative frequency of the generated $\boldsymbol{\theta}$ values are approximately those of the desired posterior distributions (Gelman et al, 1997).

The Markov chain Monte Carlo (MCMC) thus requires an initial estimate of the parameter values. Each transition distribution, $T_t\left(\boldsymbol{\theta}_t \middle| \boldsymbol{\theta}_{t-1}\right)$, is dependent only on the previous estimate of $\boldsymbol{\theta}$ and may also be dependent on the step number, t. The iterations continue until the chain is "well mixed". This means that the range of possible values of $\boldsymbol{\theta}$ has been adequately represented by the chain and the relative frequency densities approximate those of the posterior distribution. The transition distribution is dependent only on the step immediately before it, hence the Markov property (Gelman, et al, 1997). Convergence to steady state can often easily be established for relatively simple models for the transition probabilities.

Metropolis-Hastings Algorithms

The Metropolis-Hastings (Hastings, 1970) algorithm is the most general of the commonly used MCMC techniques for finding posterior distributions. Here, each sequence, $\boldsymbol{\theta}_t$, is generated so that it is a random walk at each step. The procedure starts with an initial value, $\boldsymbol{\theta}_0$, taken from some starting distribution. Given either the joint or marginal posterior distribution, $\boldsymbol{\theta}_0$ can be chosen more effectively, but many initial values will result in convergence.

For successive steps in the process, a candidate vector of parameters, $\boldsymbol{\theta}^*$, is chosen from a jumping or proposal distribution. This distribution, $J_t\left(\boldsymbol{\theta}^* \middle| \boldsymbol{\theta}_{t-1}\right)$, is dependent upon the current candidate, $\boldsymbol{\theta}_{t-1}$, for $\boldsymbol{\theta}$. The candidate point is accepted with probability

$$r = \min\left(1, \frac{p\left(\boldsymbol{\theta}^* \middle| \mathbf{y}\right) \Big/ J_t\left(\boldsymbol{\theta}^* \middle| \boldsymbol{\theta}_{t-1}\right)}{p\left(\boldsymbol{\theta}_{t-1} \middle| \mathbf{y}\right) \Big/ J_t\left(\boldsymbol{\theta}_{t-1} \middle| \boldsymbol{\theta}^*\right)}\right).$$

Notice that in the case where the jumping distribution is symmetric, that is $J_t\left(X|Y\right) = J_t\left(Y|X\right)$, the acceptance probability reduces to

$$r = \min\left(1, \frac{p\left(\boldsymbol{\theta}^* \middle| \mathbf{y}\right)}{p\left(\boldsymbol{\theta}_{t-1} \middle| \mathbf{y}\right)}\right).$$

This special case of the Metropolis-Hastings algorithm is the original "Metropolis" algorithm published by Metropolis et. al. (1953). Here it is easier to see that when the posterior density is increased by the new candidate, $\boldsymbol{\theta}^*$, it will be accepted with 100% probability. In either case, when $r < 1$, $\boldsymbol{\theta}^*$ will be accepted if a randomly selected uniform random number, *U*, is less than or equal to *r*. This is equivalent with the statement that $\boldsymbol{\theta}^*$ is accepted with the probability given above. Otherwise, $\boldsymbol{\theta}^*$ is rejected and a new candidate value needs to be sampled from the same jumping distribution.

Regardless of the choice of the proposal distribution, $J_t(\bullet)$, the Markov chain created by successively updating estimates for $\boldsymbol{\theta}$, has a stationary distribution that is the desired posterior distribution (Gilks et. al., 1996, p. 7). Despite guaranteed convergence, there are some jumping distributions that are better than others. It is best to choose a distribution that is easy to sample from. That is, for all possible $\boldsymbol{\theta}$, samples from

$J_t(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ need to be convenient. It is also necessary that the jumping distribution be chosen in such a way that the probability of acceptance, *r*, is easy to compute. The best choices of $J_t(\bullet)$ will result in quickly moving random walks with candidate values generated that are not rejected too frequently. A frequency of acceptance of 25% to 40% is recommended (Gelfand and Smith, 1990).

In many cases, the posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$, is not completely known, but is partially known modulo the normalizing constant, such as when formulating Bayes' Theorem,

$$p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta).$$

The great advantage of MCMC methods over direct sampling is that only ratios of the posterior need to be computed,

$$\frac{p(\theta^*|\mathbf{y})}{p(\theta_{t-1}|\mathbf{y})} = \frac{p(\mathbf{y}|\theta^*)p(\theta^*)}{p(\mathbf{y}|\theta_{t-1})p(\theta_{t-1})}.$$

In these ratios, the normalizing constant cancels, and is thus no longer an issue. It should be noted that for higher dimensional parameter spaces, calculating the normalizing constant proves particularly difficult because it is a challenging high-dimensional integration problem.

The Gibbs Sampler

There exist very few high-dimensional jumping distributions suitable for random draws. Thus, a big dimensional parameter vector could be challenging for formulating an

easy jumping distribution for Metropolis-Hastings. This is where the Gibbs sampler provides a framework for repeatedly updating one-dimensional parameter components.

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm, which results in 100% acceptance of proposal values. The Gibbs sampler also provides an easy scheme to deal with a parameter vector. The Gibbs sampler uses the full conditional one-dimensional posterior distribution of one parameter given all the others, $p\left(\theta_j \middle| \boldsymbol{\theta}_{-j}, \mathbf{y}\right)$. Generation of random values from this distribution and cycling through all components of the vector $\boldsymbol{\theta}$ and updating values at each step provides a Markov chain that has $p\left(\boldsymbol{\theta}|\mathbf{y}\right)$ as its steady state distribution. Block schemes, which allow for updating an entire subvector of $\boldsymbol{\theta}$ are also possible. For more detail on the Metropolis-Hastings algorithms or the Gibbs sampler, see Gilks, et. al. (1996), Gelman et al (1997) or Casella and George (1992). Recall that the CAR model defines the joint distribution of all data in terms of its full conditional distributions, and thus the Gibbs sampler arises as a natural scheme for simulation based inference.

A Bayesian Hierarchical Model

Bayesian techniques, in general, require the use of prior information combined with the likelihood of the parameters to derive posterior probability densities for each of the model parameters. When more than one level of priors and parameters are needed, a hierarchical model is developed.

Assume that $\mathbf{z}$ is a spatial process that follows the computer efficient CAR model. Let $\mathbf{y}$ be one realization of this process. Then, one can express the data vector $\mathbf{y}$ in terms of the process $\mathbf{z}$ as

$$\mathbf{y} = \mathbf{z} + \boldsymbol{\varepsilon}$$

where the vector $\boldsymbol{\varepsilon}$ is normally distributed noise centered at zero with variance σ_y^2 independent of $\mathbf{z}$. Thus, the joint distribution of the data can be written as

$$\mathbf{y} \left| \mathbf{z}, \sigma_y^2 \sim N\left(\mathbf{z}, \sigma_y^2 \mathbf{I}\right). \right.$$

To fully describe the distribution of the data, one needs to specify the distribution of both the spatial process and the variability in the data. As is common practice in Bayesian Gaussian linear models, the variance, σ_y^2, is assumed to follow an inverse gamma distribution with parameters α_y and γ_y, which is a conjugate prior. Since the spatial process is assumed to follow a Gaussian CAR model, its distribution is

$$\mathbf{z} \left| \boldsymbol{\beta}, \sigma_z^2, \phi \sim N\left(\mathbf{X}\boldsymbol{\beta}, \sigma_z^2 \mathbf{Q}^{-1}\right) \right.$$

where $\mathbf{Q}$ is the rescaled precision matrix as defined for the computer efficient CAR, $\mathbf{X}\boldsymbol{\beta}$ allows for linear trend over some set of independent explanatory variables, and ϕ is the spatial interaction term.

The introduction of additional parameters in the prior distributions, so-called hyperparameters, requires another level of priors. Here, the distribution of $\boldsymbol{\beta}$, σ_z^2, and ϕ need to be specified. A convenient hyperprior distribution for $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} \left| \boldsymbol{\beta}_0, \sigma_\beta^2 \sim N\left(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta\right). \right.$$

In most cases, $\boldsymbol{\beta}_0$ is just the zero vector as is tested against in regression analysis, although it can be chosen to be some other $p \times 1$ vector. The variance covariance matrix in this case is often simply $\sigma_\beta^2 \mathbf{I}$ where the constant σ_β^2 is usually chosen based on past experience or some other knowledge, or is noninformative as a sufficiently large value.

As with the variance constant for the data, the variance for the spatial process, σ_z^2, is assumed to follow an inverse gamma distribution with parameters α_z and γ_z. In both cases, if no strong prior information is available, the parameters for the inverse gamma distribution are chosen constants that result in relatively flat priors. The log normal distribution is chosen for the spatial interaction parameter ϕ. This implies that the values of ϕ are always positive, since to have negative spatial association implies that measurements at locations close together have opposite signs, something that tends not to occur in applications. Thus,

$$\upsilon = \log(\phi) \sim N(\mu_\phi, \sigma_\phi^2).$$

Table 2.1 summarizes the hierarchical model setup.

Estimating the parameters in a Hierarchical Model

Estimation of the parameters specified in the above hierarchical model is done by simulating values of the full conditional distributions of $\mathbf{z}, \sigma_y^2, \beta_0, \sigma_z^2$, and ν using MCMC. For large spatial areas, this requires inverting and finding the determinant of an $n \times n$ matrix, where n can be in the thousands. To remedy this problem and increase computational efficiency, an orthogonal transformation will be applied to the data that removes spatial structure. This creates for relatively quick computation since it results in a sparse precision matrix. This transformation is addressed in chapter III and will be applied before any parameter estimation is performed.

Data:	$\mathbf{y}\|\mathbf{z},\sigma_y^2 \sim N(\mathbf{z},\sigma_y^2\mathbf{I})$	$\mathbf{y}$ is the data vector $\mathbf{z}$ is the spatial process σ_y^2 is the variance of the data
Level 1:	$\mathbf{z}\|\boldsymbol{\beta},\sigma_z^2,\phi \sim N(\mathbf{X}\boldsymbol{\beta},\sigma_z^2\mathbf{Q}^{-1})$	$\boldsymbol{\beta}$ are the regression parameters σ_z^2 is the variance in the spatial process $\mathbf{Q}^{-1}$ is defined as for the computer efficient CAR model
	$\sigma_y^2 \sim InvGamma(\alpha_y,\gamma_y)$	α_y and γ_y are constant chosen parameters
Level 2:	$\boldsymbol{\beta}\|\boldsymbol{\beta}_0,\sigma_\beta^2 \sim N(\boldsymbol{\beta}_0,\boldsymbol{\Sigma}_\beta)$	$\boldsymbol{\beta}_0$ is a constant vector usually chosen to be zero $\boldsymbol{\Sigma}_\beta$ is the variance covariance matrix for $\boldsymbol{\beta}$ and is often chose to be $\sigma_\beta^2\mathbf{I}$ where σ_β^2 is a chosen constant.
	$\sigma_z^2 \sim InvGamma(\alpha_z,\gamma_z)$	α_z and γ_z are chosen constants
	$\nu = \log(\phi) \sim N(\mu_\phi,\sigma_\phi^2)$	μ_ϕ and σ_ϕ^2 are chosen constants

Table 2.1: Hierarchical spatial model setup including prior distributions and description of distribution parameters (where necessary).

CHAPTER III

COMPUTATIONAL EFFICIENCY

Overview

Despite the fact that the parameters in the CAR model proposed by Pettitt, Weir, and Hart (2001) were chosen in such a way as to make parameter estimation more computationally efficient, the fact that the data values are still correlated over space can be computationally demanding when many repeated calculations are required such as in maximum likelihood and MCMC based estimation. In this chapter we propose a transformation that results in a process that is uncorrelated over space. Thus, the process variance contains no covariance component, resulting in full conditional posterior distributions that are easier to calculate and have a simpler form. In fact, the transformed model is simply a random effects linear regression model.

An alternative and an enhancement to this data transformation for gridded data, lies in circulant embedding. By enclosing the original lattice from which the data is collected in a grid that is wrapped around a torus, all observed locations would have the same number of neighbors. This creates a weight matrix that allows for easily computing eigenvalues and eigenvectors, the most computationally taxing part of parameter estimation.

Structure Removing Data Transformation

In order to simplify calculations necessary to estimate the model parameters, the following transformation is proposed. Recall that in the computer efficient model the

precision matrix can be expressed as in (1.2). Using the weight matrix $\mathbf{G}$ and the matrix $\mathbf{D}$ defined in (1.6), $\mathbf{Q}$ can be rewritten as

$$\mathbf{Q} = \mathbf{I} - \phi(\mathbf{G} - \mathbf{D}).$$

Recall that the assumption $\phi > 0$ has been made. Then the eigenvalues of $\mathbf{Q}$ can be found through the eigenvalues of $\mathbf{G} - \mathbf{D}$ as in (1.7). The eigenvalues and eigenvectors of $\mathbf{G} - \mathbf{D}$ and the resulting eigenvalues and eigenvectors of $\mathbf{Q}$ are found using a singular value decomposition as described in chapter I.

The proposed structure removing transformation involves the eigenvalues and eigenvectors already calculated by way of $\mathbf{G} - \mathbf{D}$. Recall that the SVD results in

$$\mathbf{F}\Lambda\mathbf{F}^{\mathrm{T}} = \mathbf{G} - \mathbf{D}.$$

Note that by the properties of orthonormal vectors, $\mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{F}\mathbf{F}^{\mathrm{T}} = \mathbf{I}$. Thus, $\mathbf{F}^{\mathrm{T}} = \mathbf{F}^{-1}$ and it can be shown that the SVD of $\mathbf{Q}$ and $\mathbf{Q}^{-1}$ result in

$$\mathbf{Q} = \mathbf{F}\,diag\left(1 - \phi\lambda_i\right)\mathbf{F}^{\mathrm{T}}$$

and

$$\mathbf{Q}^{-1} = \mathbf{F}\,diag\left(\frac{1}{1 - \phi\lambda_i}\right)\mathbf{F}^{\mathrm{T}}.$$

By transforming the data and the process by way of the eigenvectors, the terms in the resulting hierarchical model are related to the original model such that $\tilde{\mathbf{y}} = \mathbf{F}^{\mathrm{T}}\mathbf{y}$, $\tilde{\mathbf{z}} = \mathbf{F}^{\mathrm{T}}\mathbf{z}$, and $\tilde{\mathbf{X}} = \mathbf{F}^{\mathrm{T}}\mathbf{X}$. Hence, the hierarchical model for the transformed data is identical to the structure in the original model. The only difference is that the $\tilde{\mathbf{z}}$ are uncorrelated with new variance matrix

$$\sigma_z^2\,diag\left(\frac{1}{1 - \phi\lambda_i}\right).$$

Recall that before the transformation, the variance covariance matrix for $\mathbf{z}$ was

$$\sigma_z^2 \mathbf{Q}^{-1}.$$

The resulting changes in the hierarchical model make for easier calculations of full conditional posteriors necessary for parameter estimation, and are summarized in Table 3.1. Parameter estimation is done using these transformed values and the back transformation, $\mathbf{F}\tilde{\mathbf{z}}$, is used after the parameter estimation is complete to obtain the original process estimates.

Full Conditional Posterior Distributions

When dealing with higher-dimensional parameter models, such as hierarchical models, it is easiest to estimate parameters using Markov chain Monte Carlo estimation procedures such as the Gibbs sampler. These procedures obviate the need for the integration of the normalizing constant. Instead, it is then necessary to compute full conditional posteriors in order to estimate parameters. That is, distributions need to be found for each parameter given all other appropriate parameters and the data. So, in the case of the spatial hierarchical model described above, the full conditional posteriors need to be found for z, σ_y^2, σ_z^2, $\boldsymbol{\beta}$, and ϕ. These are found using standard Bayesian techniques, the details of which are described in Appendix A and are summarized in Table 3.2. For additional information see Gelman, et. al. (1997). Note that the hierarchical model formulation combined with the use of conjugate priors helps to facilitate the derivations of the full conditional distributions.

Data:	$\tilde{\mathbf{y}}\left\|\mathbf{z},\sigma_y^2 \sim N\left(\mathbf{F}^{\mathrm{T}}\mathbf{z},\sigma_y^2\mathbf{I}\right)\right.$	$\tilde{\mathbf{y}}$ is the transformed data vector $\mathbf{F}^{\mathrm{T}}\mathbf{y}$ $\mathbf{z}$ is the spatial process σ_y^2 is the variance of the data
Level 1:	$\tilde{\mathbf{z}}\left\|\boldsymbol{\beta},\sigma_z^2,\phi \sim N\left(\mathbf{F}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta},\sigma_z^2 diag\left(\frac{1}{1-\phi\lambda_i}\right)\right)\right.$	$\tilde{\mathbf{z}}$ is the transformed process $\mathbf{F}^{\mathrm{T}}\mathbf{z}$ $\boldsymbol{\beta}$ are the slope parameters σ_z^2 is the variance in the spatial process
	$\sigma_y^2 \sim InvGamma(\alpha_y,\gamma_y)$	α_y and γ_y are constant chosen parameters
Level 2:	$\boldsymbol{\beta}\left\|\boldsymbol{\beta}_0,\sigma_z^2 \sim N\left(\boldsymbol{\beta}_0,\boldsymbol{\Sigma}_\beta\right)\right.$	$\boldsymbol{\beta}_0$ is a constant vector usually chose to be zero $\boldsymbol{\Sigma}_\beta = \sigma_z^2\mathbf{T}_0$ is the variance covariance matrix for $\boldsymbol{\beta}$ and is often chosen to be $\sigma_z^2\mathbf{I}$ where σ_z^2 is the process variance constant.
	$\sigma_z^2 \sim InvGamma(\alpha_z,\gamma_z)$	α_z and γ_z are chosen constants
	$\nu = \log(\phi) \sim N\left(\mu_\phi,\sigma_\phi^2\right)$	μ_ϕ and σ_ϕ^2 are chosen constants

Table 3.1: Hierarchical spatial model setup after structure removing transformation. Also includes prior distributions and description of distribution parameters (where necessary).

Parameter Estimation

Once the orthogonal transformation is applied and has removed the spatial structure, the transformed parameters can be estimated using the MCMC techniques described in Chapter II. The computational difficulties in finding the determinant and inverse of the precision matrix has been reduced significantly since the elements in the transformed process $\tilde{\mathbf{z}}$ share no spatial correlation. Since the $\tilde{\mathbf{z}}$ are uncorrelated, they can be generated simultaneously from the full conditional posterior distributions in a block instead of individually. Thus, the Markov chain works much faster and the resulting chain lengths are reduced.

The MCMC sampling requires the full conditional posterior distributions of all the parameters in the transformed hierarchical model. In general, for the parameter vector $\boldsymbol{\theta}$, the full posterior distribution is $p\left(\theta_i \middle| data, \boldsymbol{\theta}_{-i}\right)$. Here, θ_i is the ith component of the parameter vector and $\boldsymbol{\theta}_{-i}$ represents all the model parameters except for the ith. Details for determination of the full conditional posterior distributions can be found in appendix A. Since the full conditional posterior distribution for the spatial parameter ϕ can not be written in closed form, the estimation procedure involves using the Gibbs sampler with a Metropolis-Hastings step. In this case, a log scale will be applied since the proposal distribution is lognormal. Thus, $\psi = \log(\phi)$ and generated values, ψ^*, will come from the normal distribution $N\left(\psi, \sigma_\psi^2\right)$ where σ_ψ^2 is tuned for optimal rates of acceptance.

Distribution	Additional Information
$\tilde{\mathbf{z}}\left\|\tilde{\mathbf{y}},\boldsymbol{\beta},\sigma_y^2,\sigma_z^2,\phi\right. \sim$ $N\left(\tilde{\mathbf{X}}\boldsymbol{\beta}+diag\left(w_i\right)\left(\tilde{\mathbf{y}}-\tilde{\mathbf{X}}\boldsymbol{\beta}\right),\sigma_z^2 diag\left(\frac{1-w_i}{\eta_i}\right)\right)$	$\eta_i=1-\phi\lambda_i$ $w_i=\frac{\sigma_z^2}{\sigma_z^2+\sigma_y^2\eta_i}$
$\sigma_y^2\left\|\tilde{\mathbf{z}},\tilde{\mathbf{y}},\boldsymbol{\beta},\sigma_z^2,\phi\right. \sim InvGam\left(\alpha_{ypost},\gamma_{ypost}\right)$	$\alpha_{ypost}=\alpha_y+\frac{n}{2}$ $\gamma_{ypost}=\gamma_y+\frac{1}{2}\sum_{i=1}^{n}\left(y_i-z_i\right)^2$
$\sigma_z^2\left\|\tilde{\mathbf{z}},\tilde{\mathbf{y}},\boldsymbol{\beta},\sigma_y^2,\phi\right. \sim InvGam\left(\alpha_{zpost},\gamma_{zpost}\right)$	$\mathbf{e}_i=\left(e_1,\ldots,e_n\right)^T=\tilde{\mathbf{z}}-\tilde{\mathbf{X}}\boldsymbol{\beta}$ $\alpha_{zpost}=\alpha_z+\frac{n}{2}$ $\gamma_{zpost}=\gamma_z+\frac{1}{2}\sum_{i=1}^{n}\eta_i e_i^2$
$\boldsymbol{\beta}\left\|\tilde{\mathbf{z}},\tilde{\mathbf{y}},\sigma_z^2,\sigma_y^2,\phi\right. \sim$ $N\left(\mathbf{G}_\mathbf{T}\mathbf{H}^{-1}\hat{\boldsymbol{\beta}}+\left(\mathbf{I}-\mathbf{G}_\mathbf{T}\mathbf{H}^{-1}\right)\boldsymbol{\beta}_0,\sigma_z^2\left(\mathbf{I}-\mathbf{G}_\mathbf{T}\mathbf{H}^{-1}\right)\mathbf{T}_0\right)$	$\mathbf{H}^{-1}=\tilde{\mathbf{X}}^{\mathrm{T}}diag\left(\eta_i\right)\tilde{\mathbf{X}}$ $\mathbf{G}_\mathbf{T}=\left(\mathbf{H}^{-1}+\mathbf{T}_0^{-1}\right)^{-1}$ $\hat{\boldsymbol{\beta}}=\mathbf{H}\tilde{\mathbf{X}}^{\mathrm{T}}diag\left(\eta_i\right)\tilde{\mathbf{z}}$
$\phi\left\|\tilde{\mathbf{z}},\tilde{\mathbf{y}},\sigma_z^2,\sigma_y^2,\tilde{\boldsymbol{\beta}}\right.$ is not available in closed form	

Table 3.2: Full conditional posterior distributions for spatial hierarchical model parameters.

CHAPTER IV

AN EXTENDED CLASS OF SPATIAL AUTOREGRESSIVE MODELS

Overview

In Chapter I, both the CAR and SAR models, as well as the intrinsic CAR model, were described for analyzing spatial data. The conditional autoregressive (CAR) model was expanded to a computer efficient model that allows for direct and computationally fast calculation of the precision matrix.

The CAR model is somewhat limited in its application because it is best suited to modeling random fields that are not smooth in space. Besag and Kooperberg (1995) note that in cases where the spatial process is even slightly smooth, CAR models provide a poor representation of underlying patterns or the parameter estimates are on or near the boundary of the parameter space. Here, they examine the "intrinsic" CAR model (Kunsch, 1987), which is defined by a semi positive definite precision matrix and allows for the modeling of non-stationary processes. Relationships between CAR models and geostatistical models are explored by Griffith et al (1996). In particular, the rough correspondence of CAR models to exponentially decaying correlation functions, an indication of a non-differentiable random field, is noted.

The simultaneous autoregressive (SAR) model is only slightly better at explaining smoother spatial patterns than the CAR model. For even relatively smooth random fields, the best estimates of the interaction parameters for the CAR and SAR models are near the boundary. In the case of the CAR model, this results in an estimate of the spatial

parameter, ϕ, at the edge of its parameter space. In this case, $\phi \to \infty$ and the resulting model becomes the intrinsic CAR (Besag and Kooperberg, 1995).

Extension of Pettitt et al CAR Model – the Extended Autoregressive (EAR) Model

It was found that the CAR, SAR, and intrinsic CAR were not capable of fitting highly distinct smooth spatial patterns. Instead, a parameterization that extends from the computationally efficient model described by Pettitt, Weir, and Hart (2002) by incorporating a second spatial parameter is proposed to provide a smoother model of random fields. This model includes both a spatial interaction parameter, much like in the CAR and SAR models, and a parameter that relates to the smoothness of the random field. This extended class of spatial interaction model has as a special case the CAR and SAR models.

The computer efficient CAR model is extended using the eigenvalue decomposition of the standardized neighbor weight matrix, $\mathbf{G}-\mathbf{D}$. The random vector $\mathbf{z}$ is defined such that

$$\mathbf{z} \sim N(\mu, \sigma^2 \mathbf{Q}^{-1}),$$

where for some $\theta > 0$ and $\phi > 0$

$$\mathbf{Q} = \mathbf{F} diag(1-\phi\lambda_i)^{\theta} \mathbf{F}^{\mathrm{T}} \tag{4.1}$$

and

$$\mathbf{Q}^{-1} = \mathbf{F} diag\left(\frac{1}{1-\phi\lambda_i}\right)^{\theta} \mathbf{F}^{\mathrm{T}} \tag{4.2}$$

Note that like the Pettitt, et. al. parameterization ϕ is specified to be strictly larger than zero.

The addition of the parameter θ to the model governs the smoothness of the random field. As an example, a 30 by 30 grid has been created representing a spatial random field following the extended model. The values for each grid cell were generated by multiplying 30 i.i.d. standard normal variates, $\mathbf{z}_{\text{stand}}$, by the square root variance matrix,

$$\boldsymbol{\Sigma}^{1/2} = \mathbf{F} diag\left(\frac{1}{1-\phi\lambda_i}\right)^{\theta/2} \mathbf{F}^{\mathrm{T}}.$$

As can be seen in Figure 4.1, as θ increases so does the smoothness of the random field. This allows for the modeling of a spatial random field with any level of smoothness. Notice that when $\theta = 1$, the CAR model results. When $\theta = 2$, the resulting model is similar to the usual SAR model.

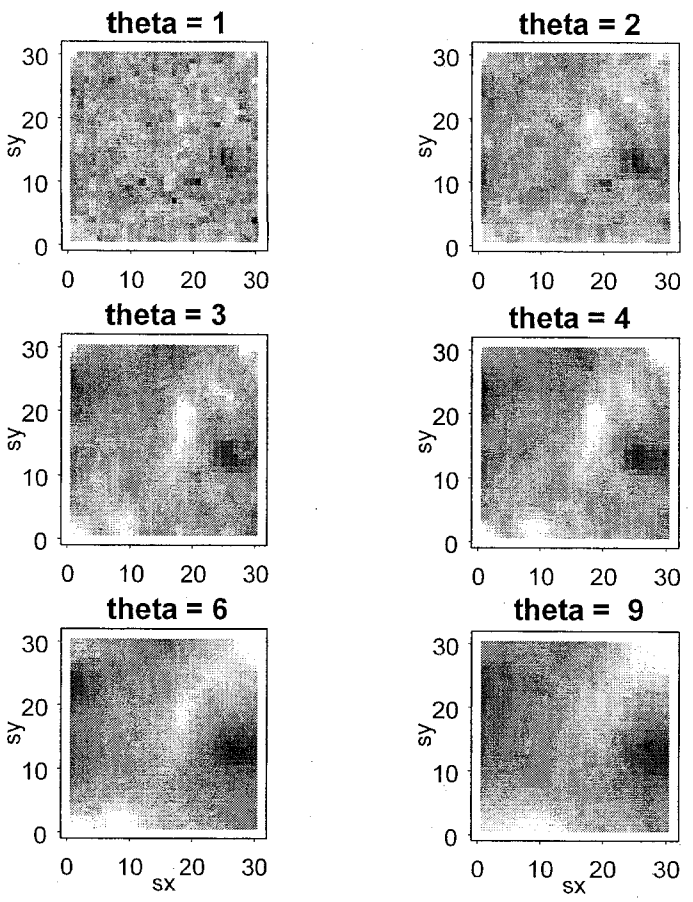

Figure 4.1: Simulated random fields with mean 0 of the extended class with $\phi = 3$ for several values of θ, and $\sigma^2 = 1$. Note the same $\mathbf{z}_{stand}$ was used for each realization.

Circulant Embedding

Consider data collected on a rectangular lattice, either regular in boundary shape or not. A slightly larger regular rectangular lattice can enclose this original lattice. Overlaying this larger lattice on a torus essentially wraps the lattice around on itself so that all the locations are laid out sequentially. Figure 4.2 below demonstrates this for an irregular grid embedded into a 7×5 lattice. The original irregular grid is shaded in gray. The locations are numbered after the embedding starting with 1 in the lower left corner and ending with 35 in the upper right. Wrapping this new grid around a torus allows for

every location at which an observation is made to have exactly the same number of neighbors and maintains that each copy of a particular location has exactly the same neighbors.

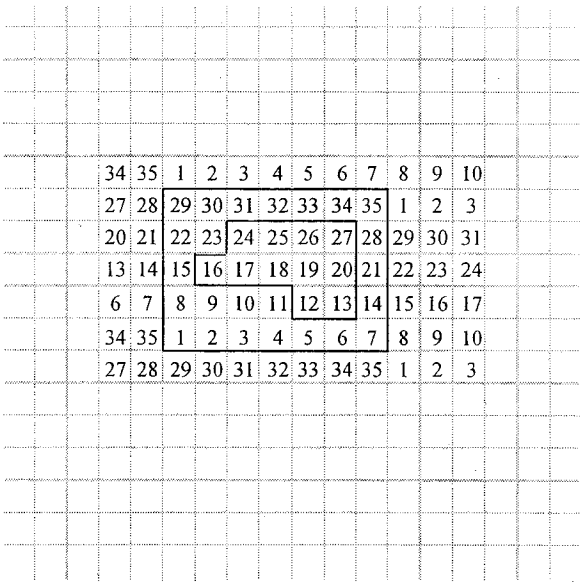

Figure 4.2: Circulant embedding demonstrated on an irregular lattice embedded in a 7×5 grid and numbered to show wrapping on a torus. The gray shaded area is the original grid and rectangle represents the larger lattice into which the original was embedded.

In the case of first order neighbors for example, every location would have exactly four neighbors. In general, suppose the larger square lattice has dimensions r rows and c columns resulting in $m = r \times c$ lattice points. Then, location 1, for example, has first order neighbors s_2, s_m, s_{m-c+1}, and s_{1+c}. From the diagram above, it is evident that the pattern will continue. Thus, this embedding scheme becomes circulant in nature. The fact that the region is expanded indicates that the original lattice on which the observations are taken are unaffected by the "additional" neighbors. Using this type of embedding creates a **G** matrix and thus a **Q** matrix that is not only sparse but also symmetric block circulant.

One nice property of symmetric circulant matrices is that the eigenvectors are constant for all matrices of the same size. This property is helpful because as the precision matrix is updated during parameter estimation, new eigenvectors do not need to be calculated. To find the eigenvectors of a $k \times k$ symmetric circulant matrix, consider the Fourier basis with dimension $n = k^2$, where n is the number of spatial locations. Then, $w_j = \frac{2\pi j}{n}$ for $j = 0,1,...,n-1$ are the angles formed by dividing the unit circle into n equal pieces. Then, for odd n, a set of orthonormal eigenvectors can be described as

$$\mathbf{c}_0 = \sqrt{\frac{1}{n}}[1,1,...,1],$$

$$\mathbf{c}_j = \sqrt{\frac{2}{n}}[1,\cos\omega j,\cos 2\omega j,...,\cos(n-1)\omega j],$$

and

$$\mathbf{s}_j = \sqrt{\frac{2}{n}}[1,\sin\omega j,\sin 2\omega j,...,\sin(n-1)\omega j].$$

The fact that n needs to be odd indicates that the embedded lattice must contain both and odd number of rows and an odd number of columns. The eigenvalues are then the real part of the fast Fourier transforms (FFT) of any row of the embedded $(\mathbf{G}\text{-}\mathbf{D})$ matrix. Then, assuming that the elements of the row represent the coefficients of a function, $m(h)$, the eigenvalues are

$$\lambda_i = m(h)e^{-i\omega_j h}.$$

Note that since the matrix is symmetric block circulant, each row contains the same elements shifted over one unit, so the FFT can be applied to any row to calculate the eigenvalues. This property is one of the basic results that will be applied for better

computational efficiency in parameter estimation for the CAR and extended autoregressive functions. For more information on circulant matrices and finding eigenvalues and eigenvectors see Brockwell and Davis (2002), Rue and Held (2005), or Wood and Chan (1994).

In order for this type of embedding to be valid for use in CAR and EAR models, it is necessary for the precision matrix, $\mathbf{Q} = \mathbf{I} - \phi(\mathbf{G} - \mathbf{D})$, to be symmetric positive definite. To demonstrate this, recall that the weight matrix $\mathbf{G}$ contains components representing the strength of the neighbor relationship for each pair of points in the lattice. For now, consider only the uniform weight system described by

$$g_{ij} = \begin{cases} 0 & s_i \notin N(s_j) \\ 1 & s_i \in N(s_j) \end{cases}. \tag{4.3}$$

For any order of neighbors considered, this results in a sparse symmetric matrix. Creating this matrix so that in addition to (4.3), $g_{1c} = 1$ and $g_{r1} = 1$, as in the circulant embedding scheme, makes $\mathbf{G}$ circulant. Then the elements of $\mathbf{D}$ represent the number of neighbors for each spatial location. Using circulant embedding, this number is constant and thus $\mathbf{D} = d\mathbf{I}$. Since $\mathbf{G}$ is symmetric and both $\mathbf{D}$ and $\mathbf{I}$ are diagonal, $\mathbf{Q} = \mathbf{I} - \phi(\mathbf{G} - \mathbf{D})$ is symmetric.

Recall the diagonal dominance property for positive definiteness as in equation (1.3). Consider first the diagonal elements, Q_{ii}. Since G has only zeros on the diagonal, $-\phi(\mathbf{G} - \mathbf{D})$ has diagonal elements equal to ϕd. By assumption, $\phi > 0$. Since d is the number of neighbors, it too is positive. When $\mathbf{I}$ is added to $-\phi(\mathbf{G} - \mathbf{D})$, one gets added

to the diagonal elements resulting in the diagonals of **Q** being $1+\phi d$. The off-diagonal elements of **I** are zero as are those of **D**. So,

$$\sum_{j:j\neq i} |Q_{ij}| = \sum_{j:j\neq i} |-\phi G_{ij}| .$$

Since uniform weights are assumed, there are d off diagonal elements that are equal to one and the rest are zero. So,

$$\sum_{j:j\neq i} |-\phi G_{ij}| = \sum_{j:j\neq i} \phi G_{ij} = \phi d .$$

Thus,

$$Q_{ii} - \sum_{j:j\neq i} |Q_{ij}| = (1+\phi d) - \phi d = 1$$

which is always greater than zero. Therefore, the precision matrix with uniform weights is positive definite.

Is an EAR Model a Markov Random Field?

By construction, the EAR model in general is not a Markov random field. That is, in this formulation, the joint distribution of one realization from the process given all other locations, $p(z_i|\mathbf{z}_{-i})$ cannot be determined using only the neighbors of that location. In this case, $p(z_i|\mathbf{z}_{-i})$ is also dependent upon the smoothing parameter θ, which is common to all locations. Recall that this is important because in order to use the Gibbs sampler for estimation, it is necessary that when values are simulated from the joint distribution, that this distribution is both stationary and unique. It will be shown in the following text that under some conditions, the EAR model is equivalent to a higher order

CAR model which is a Markov random field. Rue and Tjemeland (2002) have shown a similar correspondence between geostatistical models and Markov random fields.

To consider a higher order CAR model, it is necessary to separate **G** to distinguish the order of the neighbors being considered. Then,

$$\mathbf{G} = \mathbf{M}_1 + \mathbf{M}_2 + ... + \mathbf{M}_k,$$

where

$$M_j = \begin{cases} 1 & \text{if } s_i \text{ is a j}^{\text{th}} \text{ order neighbor} \\ 0 & \text{otherwise} \end{cases}.$$

Using the Pettitt et. al. parameterization where $\phi > 0$, $\mathbf{Q} = \mathbf{I} - \phi\mathbf{G} - \phi\mathbf{D}$. In the case where one is only interested in the first order neighbors, $\mathbf{G} = \mathbf{M}_1$, and so

$$\mathbf{Q} = \mathbf{I} - \phi_1(\mathbf{M}_1 - \mathbf{D}_1)$$

where ϕ_1 is the spatial interaction using first order neighbors and $\mathbf{D}_1$ is the number of first order neighbors each spatial location has. Similarly, considering both first and second order neighbors results in $\mathbf{G} = \mathbf{M}_1 + \mathbf{M}_2$ and

$$\mathbf{Q} = \mathbf{I} - \phi_1(\mathbf{M}_1 - \mathbf{D}_1) - \phi_2(\mathbf{M}_2 - \mathbf{D}_2).$$

It follows then that if **G** is replaced by the sum of up to the j$^{\text{th}}$ order neighbor matrices then

$$\mathbf{G} = \mathbf{M}_1 + \mathbf{M}_2 + ... + \mathbf{M}_j$$

and

$$\mathbf{Q} = \mathbf{I} - \phi_1(\mathbf{M}_1 - \mathbf{D}_1) - \phi_2(\mathbf{M}_2 - \mathbf{D}_2) - ... - \phi_j(\mathbf{M}_j - \mathbf{D}_j). \quad (4.4)$$

The fact that **G** is symmetric circulant and **D** is a diagonal matrix, the quick computation of eigenvalues and eigenvectors using Fast Fourier Transforms (FFT) as

described above can be used on $\mathbf{G}-\mathbf{D}$. Let $\lambda_i = EV(\mathbf{G}-\mathbf{D})$. Assuming that $\mathbf{G}$ and $\mathbf{D}$ are defined as above, $\lambda_i = EV(\mathbf{G}-d\mathbf{I}) = \mu_i - d$ for $\mu_i = EV(\mathbf{G})$. Because all $\mathbf{M}_j$ are circulant and $\mathbf{D}_j = d_j\mathbf{I}$

$$\lambda_i = \phi_1(\mu_{1,i}-d_1)+\phi_2(\mu_{2,i}-d_2)+\ldots+\phi(\mu_{k,i}-d_k) \tag{4.5}$$

or

$$\mathbf{F}^{\mathrm{T}}\mathbf{Q}\mathbf{F} = diag\left[1-\phi_1(\mu_{1,i}-d_1)-\phi_2(\mu_{2,i}-d_2)-\ldots-\phi_k(\mu_{k,i}-d_k)\right]. \tag{4.6}$$

To demonstrate the correspondence between the EAR model and a higher order Markov random field, consider the first order EAR model with precision matrix

$$\mathbf{Q} = \mathbf{F}diag(1-\phi\lambda_i)^{\theta}\mathbf{F}^{\mathrm{T}}.$$

Assuming a normal distribution,

$$\mathbf{z} \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma} = \mathbf{Q}^{-1})$$

and

$$\mathbf{F}^{\mathrm{T}}\mathbf{z} \sim MVN\left(\mathbf{F}^{\mathrm{T}}\boldsymbol{\mu}, diag\frac{1}{(1-\phi\lambda_i)^{\theta}}\right).$$

Recall that any function can be approximated by a Taylor series expansion. That is, any function can be written as

$$f(x) = f(a)+f'(a)(x-a)+f''(a)\frac{(x-a)^2}{2!}+f'''(a)\frac{(x-a)^3}{3!}+\ldots$$

where a is any constant. Thus, with $a=0$

$$(1-\phi\lambda_i)^{\theta} = h(\lambda_i) = h(0)+h'(0)\lambda_i+h''(0)\frac{\lambda_i^2}{2!}+h'''(0)\frac{\lambda_i^3}{3!}+\ldots.$$

Using basic calculus,

$$h'(\lambda_i) = \theta(1-\phi\lambda_i)^{\theta-1}(-\phi)$$

and

$$h''(\lambda_i) = \theta(\theta-1)(1-\phi\lambda_i)^{\theta-1}\phi^2.$$

In general

$$h^{(p)}(\lambda_i) = \theta(\theta-1)(\theta-2)...(\theta-p+1)(1-\phi\lambda_i)^{\theta-p}(-\phi)^p$$

and thus

$$h^{(p)}(0) = \theta(\theta-1)(\theta-2)...(\theta-p+1)(-\phi)^p.$$

Substituting this into the Taylor series expansion results in

$$(1-\phi\lambda_i)^{\theta} = 1-\phi\theta\lambda_i+\phi^2\theta(\theta-1)\frac{\lambda_i^2}{2}-\phi^3\theta(\theta-1)(\theta-2)\frac{\lambda_i^3}{6}+....$$

Simplified, this becomes

$$(1-\phi\lambda_i)^{\theta} = 1-\phi\lambda_i\theta+(\phi\lambda_i)^2\frac{\theta(\theta-1)}{2}-(\phi\lambda_i)^3\frac{\theta(\theta-1)(\theta-2)}{6}+.... \quad (4.7)$$

If θ is restricted to positive integers, then the Taylor series expansion above in (4.7) terminates at the $\theta+1^{st}$ term because $h^{(\theta)}(\lambda_i) = 0$.

It can be shown in general that

$$\mathbf{M}_1^2 = 4\mathbf{I}+2\mathbf{M}_2+\mathbf{M}_3, \quad (4.8)$$

$$\mathbf{M}_1^3 = 9\mathbf{M}_2+3\mathbf{M}_4+\mathbf{M}_8, \quad (4.9)$$

etc. For example, consider the nine spatial locations in a 3×3 regular lattice. In this case, the first order neighbors of location 1 are locations 2, 4, 7, and 9 regardless of which "1" you look at. The first order neighbor weight matrix is

$$\mathbf{M}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Notice that this matrix is both symmetric and circulant. Similarly, $\mathbf{M}_2$ and $\mathbf{M}_3$ can be found to be

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$\mathbf{M}_3 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

It is easily verified that

$$\mathbf{M}_1^2 = \begin{bmatrix} 4 & 0 & 3 & 1 & 2 & 2 & 1 & 3 & 0 \\ 0 & 4 & 0 & 3 & 1 & 2 & 2 & 1 & 3 \\ 3 & 0 & 4 & 0 & 3 & 1 & 2 & 2 & 1 \\ 1 & 3 & 0 & 4 & 0 & 3 & 1 & 2 & 2 \\ 2 & 1 & 3 & 0 & 4 & 0 & 3 & 1 & 2 \\ 2 & 2 & 1 & 3 & 0 & 4 & 0 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 4 & 0 & 3 \\ 3 & 1 & 2 & 2 & 1 & 3 & 0 & 4 & 0 \\ 0 & 3 & 1 & 2 & 2 & 0 & 3 & 0 & 4 \end{bmatrix}$$

and that

$$\mathbf{M}_1^2 = 4\mathbf{I} + 2\mathbf{M}_2 + \mathbf{M}_3 .$$

The example above demonstrates that powers of first order neighbor structure can be written as the linear combination of higher order neighbor weight matrices. Thus, the polynomial in λ_i in (4.7) can be rewritten using linear combinations of eigenvalues from higher order neighbor weight structure. Since for the EAR model

$\mathbf{F}^{\mathrm{T}}\mathbf{Q}\mathbf{F} = diag\left[\left(1-\phi\lambda_i\right)^{\theta}\right]$, from equation (4.7) above

$$\mathbf{F}^{\mathrm{T}}\mathbf{Q}\mathbf{F} = diag\left[1-\phi\lambda_i\theta+\left(\phi\lambda_i\right)^2\frac{\theta(\theta-1)}{2}-\left(\phi\lambda_i\right)^3\frac{\theta(\theta-1)(\theta-2)}{6}+\ldots\right]. \qquad (4.10)$$

Then from (4.8) and (4.9)

$$\lambda_i = \mu_{1,i} - d_1$$

and

$$\lambda_i^2 = 4 + 2\mu_{2,i} + \mu_{3,i} - d_1,$$

etc. Thus, these can be substituted into (4.10) above to get a linear combination of the eigenvalues of higher order CAR neighbor structures as in (4.6). Therefore, a first order EAR model can be rewritten as a CAR model implying that it is a Markov random field.

CHAPTER V

AUTOREGRESSIVE MODELS FOR TIME SERIES DATA

Overview

The proposed model for analyzing spatio-temporal data includes, in addition to a linear trend parameter, parameters that explain temporal correlation in the residuals. Most often correlations over time are posed as an autoregressive model of order one which are widely used in time series analysis and are commonly denoted by $AR(1)$. The CAR and SAR first order formulations described for spatial data are identical to those used for time series data.

Consider data x_t, $t = 1,2,...,T$ collected over time. One approach in modeling this type of data is to use an autoregressive model of order p, or an $AR(p)$ model. In this case it is assumed that

$$x_t - \mu = \rho_1(x_{t-1} - \mu) + ... + \rho_p(x_t - \mu) + Z_t$$

where μ is the mean of the data, the ρ_j are the temporal parameters such that

$$\rho_1 = \text{corr}(y_{it}, y_{i,t+1} \mid \mathbf{y}_{-it}, \mathbf{y}_{-i,t+1}),$$

etcetera, for $-1 < \rho_j < 1$, and Z_t represents random error for the process at time t. Equivalently, if the data is centered with mean zero,

$$x_t = \rho_1 x_{t-1} + ... + \rho_p x_{t-p} + Z_t$$

Usually, it is assumed that the error is Gaussian, or that $Z_t \sim N(0, \sigma_z^2)$.

First Order Autoregressive Models – AR(1)

In the case where one is only concerned with the immediate preceding time step to model the data at time t, $p=1$. The model reduces to

$$x_t - \mu = \rho(x_{t-1} - \mu) + Z_t$$

or

$$x_t = \rho_1 x_{t-1} + Z_t .$$

Using back step notation, it can be shown that

$$(1-\rho B)x_t = Z_t$$

where B denotes the backshift operator such that $Bx_t = x_{t-1}$.

Solving for x_t and incorporating all previous steps, it can be shown that

$$x_t = Z_t + \rho Z_{t-1} + \rho^2 Z_{t-2} + \ldots .$$

Since all the Zs have the same distribution,

$$E(X_t) = 0$$

and

$$\sigma_x^2 = Var(X_t) = \sigma_z^2(1+\rho+\rho^2+\ldots) .$$

Letting $\mathbf{X} = (x_1, x_2, \ldots, x_T)^T$ be the vector containing all the data, the joint distribution can be written as

$$\mathbf{X} \sim N(\mathbf{0}, \sigma_z^2 \mathbf{Q}_\mathbf{T}^{-1})$$

where

$$\mathbf{Q}_{\mathrm{T}}^{-1}=\begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \cdots & \\ \rho & 1 & \rho & \rho^2 & \cdots & \\ \rho^2 & \rho & 1 & \rho & \cdots & \\ \rho^3 & \rho^2 & \rho & 1 & \cdots & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \rho \\ & & & & \rho & 1 \end{pmatrix}$$

This results in a tridiagonal matrix for $\mathbf{Q}_{\mathrm{T}}$,

$$\mathbf{Q}_{\mathrm{T}}=\begin{pmatrix} c+\delta & \gamma & & & 0 \\ \gamma & c & \gamma & & \\ & \gamma & \ddots & \ddots & \\ & & \ddots & c & \gamma \\ 0 & & & \gamma & c+\delta \end{pmatrix},$$

which can be rewritten in terms of ρ as

$$\mathbf{Q}_T=\frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho & 0 & \cdots & 0 \\ -\rho & 1+\rho^2 & -\rho & \cdots & 0 \\ 0 & -\rho & 1+\rho^2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\rho \\ 0 & 0 & \cdots & -\rho & 1 \end{bmatrix}$$

$$=\frac{1+\rho^2}{1-\rho^2}\begin{bmatrix} 1/(1+\rho^2) & -\rho/(1+\rho^2) & 0 & \cdots & 0 \\ -\rho/(1+\rho^2) & 1 & -\rho/(1+\rho^2) & \cdots & 0 \\ 0 & -\rho/(1+\rho^2) & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\rho/(1+\rho^2) \\ 0 & 0 & \cdots & -\rho/(1+\rho^2) & 1/(1+\rho^2) \end{bmatrix}$$

The precision matrix can then be approximated by ignoring the "edge" effect. This results in

$$\mathbf{Q}_T(\rho)=\frac{1+\rho^2}{1-\rho^2}\left(\mathbf{I}_T-\frac{\rho}{1+\rho^2}\mathbf{V}\right)$$

where

$$\mathbf{V} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}.$$

Typical annual time series are often modeled very well by an $AR(1)$ process. This is because in most cases, the process is only related to what happened in the previous year. Since it is necessary to decide on a particular temporal autoregressive process for the spatial temporal model discussed later, only the first order autoregressive model will be considered.

CHAPTER VI

A HIERARCHICAL MODEL FOR DATA IN SPACE AND TIME

Overview

It is often of interest to take spatial models one step further to incorporate temporal trends. This type of model is used when repeated measurements representing time steps are collected at each of the spatial locations. In this case, the linear trend parameters $\boldsymbol{\beta}$ and the spatial interaction parameter ϕ as defined for the spatial model above and the parameter ρ defined for the temporal model, are all incorporated in the distribution of the data to represent linear temporal trend, spatial interaction, and temporal autoregressive behavior ($AR(1)$) respectively.

To model data collected over both space and time, the computer efficient CAR model of Pettitt, Weir, and Hart (2001) described for analyzing spatial data is modified to incorporate not only spatial interaction, but temporal dependencies as well, as in the space-time hierarchical model of Wikle, Berliner, and Cressie (1998). In this model, it is assumed that the data at each location comes from a normal distribution with errors that could contain spatial or temporal correlations. If the data do not come from a normal distribution, an appropriate transformation can be made so that the transformed data is Gaussian. The purpose here is to estimate simultaneously linear temporal trend as well as spatial and temporal structure in the residuals, or alternatively spatial structure in individual time trends, also called "spatially varying coefficients model."

A General Class of Hierarchical Gaussian Space-Time Models

Recall that the hierarchical model used in the spatial setup is defined in such a way that the theory of Markov random fields is used. So, as in the spatial model, our data $\mathbf{y}$ is written as centered at some assumed and unknown random field with error. That is, $\mathbf{y} = \mathbf{z} + \boldsymbol{\varepsilon}$ where the error terms are normally distributed with some constant variance, σ_y^2. Since the data is assumed to follow a normal distribution, the distribution of the data given the random field is

$$\mathbf{y} \,|\, \mathbf{z} \sim N\left(\mathbf{z}, \sigma_y^2 \mathbf{I}_{nT}\right).$$

Here, n is the number of spatial locations and T is the number of time points.

Unlike in the hierarchical setup used for the spatial model, the random field now contains both spatial and temporal structure. The mean of the process is the sum of both fixed and random linear effects in space or time. Let $\boldsymbol{\eta}_{st}$, $\boldsymbol{\eta}_s$, and $\boldsymbol{\eta}_t$ represent the fixed effects. These vectors are of length p_{st}, p_s, and p_t, respectively. Let $\boldsymbol{\beta}_k$ represent the k^{th} random effect in either space or time. Note that random effects cannot be in space-time because they would not be identifiable. So, the mean of the process is written

$$\mathbf{X}_{st}\boldsymbol{\eta}_{st} + \mathbf{X}_t\boldsymbol{\eta}_t + \mathbf{X}_s\boldsymbol{\eta}_s + \sum_{k=1}^{m} \mathbf{X}_k^*\boldsymbol{\beta}_k .$$

Here,

$$\mathbf{X}_s = \begin{pmatrix} \mathbf{U}_s \\ \mathbf{U}_s \\ \vdots \\ \mathbf{U}_s \end{pmatrix} = \mathbf{U}_s \otimes \mathbf{1}_T$$

is an $N \times p_s$ matrix where $N = nt$ and $\mathbf{U_s}$ is an $n \times p_s$ matrix defined by the spatial regressors. Similarly,

$$\mathbf{X}_t = \mathbf{U}_\mathbf{T} \otimes \mathbf{1}.$$

It is often the case that $\mathbf{X_{st}}$ is just $\mathbf{1}$ so that $\boldsymbol{\eta}_{st}$ is a scalar intercept term.

For each of the space varying random effects, the effects correspond to n distinct time trends, one for each spatial location. Here,

$$\boldsymbol{\beta}_\mathbf{k} = \begin{pmatrix} \beta_{k1} \\ \beta_{k2} \\ \vdots \\ \beta_{kn} \end{pmatrix}$$

are n trends over time. Then,

$$\mathbf{X}_\mathbf{k}^* = \begin{pmatrix} v_{k1}\mathbf{I_n} \\ v_{k2}\mathbf{I_n} \\ \vdots \\ v_{kT}\mathbf{I_n} \end{pmatrix} = \mathbf{v}_\mathbf{k} \otimes \mathbf{I}_\mathbf{n}$$

Where, $v_{k1}, v_{k2}, ..., v_{kT}$ are constants defining a regressor variable in the time domain. In the case of time-varying random effects, $\boldsymbol{\beta}_\mathbf{k}$ varies for each time point. Thus,

$$\mathbf{X}_\mathbf{k}^* = \mathbf{I}_\mathbf{T} \otimes \mathbf{w}_\mathbf{k}$$

where $\mathbf{w_k}$ is a vector of constants, defining a regressor variable in the space domain.

The distribution of the process is then

$$\mathbf{z} \left| \sigma_z^2, \phi, \rho \sim N\left(\mathbf{X_{st}}\boldsymbol{\eta}_\mathbf{st} + \mathbf{X_t}\boldsymbol{\eta}_\mathbf{t} + \mathbf{X_s}\boldsymbol{\eta}_\mathbf{s} + \sum_{k=1}^{m} \mathbf{X_k}\boldsymbol{\beta}_\mathbf{k}, \sigma_z^2 \mathbf{Q}_\mathbf{st}^{-1} \right) \right.$$

with $\mathbf{Q}_\mathbf{st}^{-1}$ defined by ϕ , ρ or no parameter depending on whether the random effects are space-varying, time-varying, or both. In the case of no random effect, a general space-

time covariance matrix, $\sigma_z^2 \mathbf{Q}_{sr}^{-1}$ is assumed. To insure computational efficiency, it is assumed that the space-time correlation structure is separable. That is, one assumes that the precision matrix can be written

$$\mathbf{Q}_{st} = \mathbf{Q}_t \otimes \mathbf{Q}_s$$

where $\mathbf{Q}_s$ is precision based on spatial dependencies and $\mathbf{Q}_t$ is precision based on temporal dependencies. Note that $\mathbf{Q}_s$ and $\mathbf{Q}_t$ can be as defined for the computer efficient spatial CAR model and the temporal $AR(1)$ model, or be simply diagonal matrices indicating the lack of spatial or temporal structure in the residuals. The simplest case is when both spatial and temporal random effects are incorporated in the mean. Here, spatial and temporal structure has already been explained by the model and assuming no more structure in the errors. Thus, the precision matrix is then simply

$$\mathbf{Q}_{st} = \mathbf{Q}_{tdiag} \otimes \mathbf{Q}_{sdiag},$$

a purely diagonal matrix resulting from having neither spatial nor temporal structure in the covariance.

When the mean contains space varying temporal random effects, $\mathbf{Q}_{st}$ is proposed with no spatial structure. So,

$$\mathbf{Q}_{st} = \mathbf{Q}_t \otimes \mathbf{Q}_{sdiag}$$

where $\mathbf{Q}_{sdiag}$ is a diagonal matrix indicating no spatial influence on the covariance. Similarly, in the case where the mean contains time varying spatial random effects, it is assumed that there is no temporal structure left in the errors and the precision matrix is

$$\mathbf{Q}_{st} = \mathbf{Q}_{tdiag} \otimes \mathbf{Q}_s.$$

Notice that the variance covariance matrix will depend on ϕ, ρ, or both depending on the presence of only time-varying random effects, only space-varying random effects, or both spatial and temporal random effects respectively. Thus, the prior distributions need to be specified for these as well as σ_y^2 and σ_z^2 to complete the hierarchical model. As in the spatial hierarchical model, both variance parameters are conventionally assumed to have an inverse Gamma distribution. So, the prior distribution for the measurement error is

$$\sigma_y^2 \sim InvGamma\left(\alpha_{yst}, \gamma_{yst}\right),$$

and the prior distribution for the error in the process is

$$\sigma_z^2 \sim InvGamma\left(\alpha_{zst}, \gamma_{zst}\right).$$

As in the spatial hierarchical model, the inverse gamma parameters are chosen constants.

Recall that ρ is the $AR(1)$ parameter representing temporal structure. It is assumed that $-1 \le \rho \le 1$ in temporal autoregressive processes (Chatfield, 1996). Since it is unknown which values of ρ are more likely, the prior distribution for ρ is assumed to be uniform(-1,1). As in the spatial hierarchical setup, the distribution of ϕ is chosen to be lognormal. That is

$$\nu = \log(\phi) \sim N\left(\mu_\phi, \sigma_\phi^2\right)$$

where μ_ϕ and σ_ϕ^2 are chosen constants.

As in the spatial model, the priors for each of the slope parameters are normal distributions. In the case of the fixed effects, the priors for the spatial, temporal, and space-time fixed effects are

$$\boldsymbol{\alpha}_s \sim N\left(\boldsymbol{\alpha}_{s0}, \sigma_z^2 \boldsymbol{\Sigma}_{\alpha_s}\right),$$

$$\boldsymbol{\alpha}_t \sim N\left(\boldsymbol{\alpha}_{t0}, \sigma_z^2 \boldsymbol{\Sigma}_{\alpha_t}\right),$$

and

$$\boldsymbol{\alpha}_{st} \sim N\left(\boldsymbol{\alpha}_{st0}, \sigma_z^2 \boldsymbol{\Sigma}_{\alpha_{st}}\right),$$

respectively. Notice that for all three fixed effects, the variance constant is the same as for the process, **z**. The variance-covariance matrix for each fixed effect is chosen and has no hyperprior. In many cases, the mean vector is chosen to be the zero vector.

The prior distributions, although normal, are very different for the random effects. For all random effects, the mean for the prior is **0**.

A Special Case of Space-Time Hierarchical Gaussian Models

Of particular interest is the single space-varying random effect representing the trend over time for each location. Let $\boldsymbol{\beta}$ be this random effect. The prior for this effect has distribution

$$\boldsymbol{\beta} \sim N\left(\mathbf{0}, \sigma_\beta^2 \mathbf{Q}_\beta^{-1}\right)$$

where

$$\mathbf{Q}_\beta = \mathbf{I} - \phi_\beta \left(\mathbf{G} - \mathbf{D}\right).$$

As in the computer efficient CAR model, **G** is the neighbor weight matrix and **D** is the diagonal matrix containing the row sums of **G**.

Recall that in the spatial computer efficient CAR model, the prior for the spatial interaction parameter is chosen to be a lognormal distribution. The same prior can be used for the distribution of the parameter ϕ_β.

Data:	$\mathbf{y}\|\mathbf{z},\sigma_y^2 \sim N\left(\mathbf{z},\sigma_y^2\mathbf{I}_{nT}\right)$	$\mathbf{y}$ is the data vector $\mathbf{z}$ is the underlying process σ_y^2 is the constant measurement error
Level 1:	$\mathbf{z}\|\boldsymbol{\beta},\sigma_z^2,\phi,\rho \sim N\left(\mathbf{X}\boldsymbol{\beta},\sigma_z^2\mathbf{Q}_{st}^{-1}\right)$	$\mathbf{z}$ is the underlying process $\boldsymbol{\beta}$ is the random effect representing space-varying temporal trend σ_z^2 is the variance in the spatial process $\mathbf{Q}_{st} = \mathbf{Q}_t \otimes \mathbf{I}_n$ is the precision matrix
	$\sigma_y^2 \sim InvGamma(\alpha_{yst},\gamma_{yst})$	α_{yst} and γ_{yst} are constant chosen parameters
Level 2:	$\boldsymbol{\beta}\|\sigma_\beta^2,\phi_\beta \sim N\left(\mathbf{0},\sigma_\beta^2\mathbf{Q}_\beta^{-1}\right)$	$\mathbf{Q}_\beta = \mathbf{I} - \phi_\beta(\mathbf{G}-\mathbf{D})$ where $\mathbf{G}$ is the weight matrix and $\mathbf{D}$ is the diagonal matrix containing the row sums of $\mathbf{G}$. σ_β^2 is the chosen variance constant.
	$\sigma_z^2 \sim InvGamma(\alpha_{zst},\gamma_{zst})$	α_{zst} and γ_{zst} are chosen constants
	$\nu = \log(\phi) \sim N\left(\mu_\phi,\sigma_\phi^2\right)$	μ_ϕ and σ_ϕ^2 are chosen constants
	$\rho \sim Unif(-1,1)$	All values for ρ are equally likely

Table 6.1: Spatio-temporal hierarchical setup for observations with only simple linear temporal trend including prior distributions and description of distribution parameters.

The complete hierarchical setup is summarized in table 6.1 for the case where there is only the space-varying random effect representing simple temporal trend. Notice that

when the space-varying random effect is in the model, the space-varying fixed effect is omitted as to not over parameterize the model.

Detailed Formulation for a Model with Spatially Varying Temporal Trends

As with the spatial hierarchical model, spatial structure can be removed using a transformation involving the singular value decomposition (SVD) of the $\mathbf{G}-\mathbf{D}$ within the precision matrix. Recall that in its most complex form, the spatial portion of the space-time precision matrix can be written

$$\mathbf{Q}_s = \mathbf{I} - \phi(\mathbf{G}-\mathbf{D})$$

where $\mathbf{G}$ is the weight matrix, $\mathbf{D}$ is the diagonal matrix containing the row sums of $\mathbf{G}$, and $\mathbf{I}$ is the n-dimensional identity matrix. If $\mathbf{F}$ is the matrix containing the eigenvectors of $\mathbf{G}-\mathbf{D}$ and Λ is the diagonal matrix containing the eigenvalues of $\mathbf{G}-\mathbf{D}$, then the spatial precision matrix can be rewritten as

$$\mathbf{Q}_s = \mathbf{F}\,diag\left(1-\phi\lambda_i\right)\mathbf{F}^{\mathrm{T}} \tag{6.1}$$

and the process variance-covariance structure is

$$\sigma_z^2\mathbf{Q}_s = \sigma_z^2\mathbf{F}\,diag\left(\frac{1}{1-\phi\lambda_i}\right)\mathbf{F}^{\mathrm{T}}.$$

Since the purpose of the transformation is to remove spatial structure, it need not be applied to the temporal precision matrix $\mathbf{Q}_t$.

To apply the transformation to the vector of observations, $\mathbf{y}$, recall that in the space-time framework,

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_T \end{pmatrix}.$$

Applying the transformation to each component of the data vector results in

$$\mathbf{y}_{new} = \begin{pmatrix} \mathbf{F}^T\mathbf{y}_1 \\ \mathbf{F}^T\mathbf{y}_2 \\ \vdots \\ \mathbf{F}^T\mathbf{y}_T \end{pmatrix} = \left(\mathbf{I}_T \otimes \mathbf{F}^T\right) \cdot \mathbf{y}.$$

Similarly, the transformation applied to the process results in

$$\mathbf{z}_{new} = \begin{pmatrix} \mathbf{F}^T\mathbf{z}_1 \\ \mathbf{F}^T\mathbf{z}_2 \\ \vdots \\ \mathbf{F}^T\mathbf{z}_T \end{pmatrix} = \left(\mathbf{I}_T \otimes \mathbf{F}^T\right) \cdot \mathbf{z}.$$

The measurement error then becomes

$$\left(\mathbf{I}_T \otimes \mathbf{F}^T\right)\sigma_z^2\left(\mathbf{I}_T \otimes \mathbf{F}^T\right)^T$$

which reduces to

$$\sigma_z^2\left(\mathbf{I}_T \otimes \mathbf{F}^T\mathbf{F}\right) = \sigma_z^2\mathbf{I}_{nT}.$$

To transform the distribution of the process, it is easiest if the mean is separated into its two parts. For the fixed effects, let

$$\mathbf{X} = \left[\mathbf{X}_{st} \mid \mathbf{X}_s \mid \mathbf{X}_t\right] = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{pmatrix}$$

and

$$\beta = \begin{pmatrix} \beta_{st} \\ \beta_s \\ \beta_t \end{pmatrix}.$$

Applying the transformation results in

$$\left(\mathbf{I}_T \otimes \mathbf{F}^T\right)\mathbf{X}\beta .$$

Notice that the regressors are transformed and the fixed effects remain untouched.

By applying the transformation to the space varying random effects and their corresponding regressor matrices, the result is

$$\sum_{k=1}^{m}\left(\mathbf{I}_T \otimes \mathbf{F}^T\right)\mathbf{X}_k\beta_k .$$

Recall that

$$\mathbf{X}_k = \left(\mathbf{v}_k \otimes \mathbf{I}_n\right).$$

Then, for each term in the sum

$$\left(\mathbf{I}_T \otimes \mathbf{F}^T\right)\mathbf{X}_k = \left(\mathbf{I}_T \otimes \mathbf{F}^T\right)\left(\mathbf{v}_k \otimes \mathbf{I}_n\right)$$

which simplifies to

$$\mathbf{v}_k \otimes \mathbf{F}^T .$$

Multiplying this with β_k yields

$$\left(\mathbf{v}_k \otimes \mathbf{F}^T\right)\beta_k = \begin{pmatrix} \mathbf{v}_1\mathbf{F}^T \\ \mathbf{v}_2\mathbf{F}^T \\ \\ \mathbf{v}_T\mathbf{F}^T \end{pmatrix}\beta_k = \left(\mathbf{v}_k \otimes \mathbf{I}_n\right)\mathbf{F}^T\beta_k = \mathbf{X}_k\mathbf{F}^T\beta_k .$$

Thus, for the random effects, the regressors remain unchanged while the effects are transformed as

$$\tilde{\beta}_k = \mathbf{F}^T\beta_k .$$

Then $\boldsymbol{\beta}_k \sim N\left(\mathbf{0}, \sigma_\beta^2 \mathbf{F}^T \mathbf{Q}_\beta^{-1} \mathbf{F}\right)$ where $\mathbf{F}^T \mathbf{Q}_\beta^{-1} \mathbf{F}$ is a diagonal matrix.

Applying the transformation to the time varying random effects results in

$$\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{w}_\mathrm{k}\right) = \left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\mathbf{w}_\mathrm{k}\right)$$

for each term in the sum. Multiplying by $\boldsymbol{\beta}_\mathrm{k}$ yields

$$\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\mathbf{w}_\mathrm{k}\right)\boldsymbol{\beta}_\mathrm{k}$$

which can be reduced no further. Thus, the effects remain unchanged while the regressor matrix is transformed. For information on Kronecker products, see *Kronecker Products and Matrix Calculus with Applications* (Graham, 1981).

Applying the transformation to the process precision matrix results in

$$\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)\mathbf{Q}_\mathrm{st}\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)^\mathrm{T}.$$

In the case where

$$\mathbf{Q}_\mathrm{st} = \mathbf{Q}_\mathrm{t} \otimes \mathbf{Q}_\mathrm{s}$$

the transformed precision is

$$\begin{aligned}\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)\mathbf{Q}_\mathrm{st}\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)^\mathrm{T} &= \left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\right)\left(\mathbf{Q}_\mathrm{t} \otimes \mathbf{Q}_\mathrm{s}\right)\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}\right) \\ &= \left(\mathbf{I}_\mathrm{T}\mathbf{Q}_\mathrm{t} \otimes \mathbf{F}^\mathrm{T}\mathbf{Q}_\mathrm{s}\right)\left(\mathbf{I}_\mathrm{T} \otimes \mathbf{F}\right) \\ &= \mathbf{I}_\mathrm{T}\mathbf{Q}_\mathrm{t}\mathbf{I}_\mathrm{T} \otimes \mathbf{F}^\mathrm{T}\mathbf{Q}_\mathrm{s}\mathbf{F} \\ &= \mathbf{Q}_\mathrm{t} \otimes \mathbf{F}^\mathrm{T}\mathbf{Q}_\mathrm{s}\mathbf{F}.\end{aligned}$$

With $\mathbf{Q}_\mathrm{s}$ as in equation (6.1), the transformed space-time precision matrix becomes

$$\mathbf{Q}_\mathrm{t} \otimes diag\left(1 - \phi\lambda_i\right).$$

Similarly, the transformed precision matrix for the remaining cases can be simplified and are summarized in table 6.2.

SPACE-TIME PRECISION MATRIX	TRANSFORMED SPACE-TIME PRECISION MATRIX
$\mathbf{Q}_{st} = \mathbf{Q}_t \otimes \mathbf{Q}_s$	$\mathbf{Q}_t \otimes diag(1-\phi\lambda_i)$
$\mathbf{Q}_{st} = \mathbf{Q}_t \otimes \mathbf{I}_n$	$\mathbf{Q}_t \otimes \mathbf{I}_n$
$\mathbf{Q}_{st} = \mathbf{Q}_{tdiag} \otimes \mathbf{Q}_s$	$\mathbf{Q}_t \otimes diag(1-\phi\lambda_i)$
$\mathbf{Q}_{st} = \mathbf{Q}_{tdiag} \otimes \mathbf{I}_n$	$\mathbf{Q}_{tdiag} \otimes \mathbf{I}_n$

Table 6. 2: Transformed Space-Time precision matrices using the SVD based structure-removing transformation for removing spatial structure.

Addition of the parameter θ to the Space-Time Model

Since the random effects in the space-time model parallel the purely spatial model discussed above, it is natural to apply the extended class of spatial autoregressive models to the spatio-temporal situation. This will allow the space-time model to account for smoother spatial patterns without the added burden of several extra parameters as in the CAR model with higher order neighbor structure.

Instead of assuming that the data follow the CAR model of Pettitt, et. al., the EAR model can be used to represent spatial dependencies, thus adding θ to the parameters in the space-time model. Then the space-varying random effects would be modeled as

$$\boldsymbol{\beta}\left|\sigma_\beta^2, \phi_\beta\theta \sim N\left(\mathbf{0}, \sigma_\beta^2\mathbf{Q}_\beta^{-1}\right)\right.$$

where, $\mathbf{Q}_\beta$ is the precision matrix from the EAR model and can be rewritten based on its singular value decomposition as

$$\mathbf{Q}_{\beta} = \mathbf{F} diag\left(1-\phi\lambda_i\right)^{\theta} \mathbf{F}^{\mathrm{T}}.$$

Here, the prior distribution for θ is in level three of the hierarchical model and is assumed to be log normal, or

$$\log(\theta) \sim N\left(\mu_{\theta}, \sigma_{\theta}^2\right).$$

All other parameters in the space-time hierarchical model and their priors remain the same. When the structure removing transformation is applied the spatial random effects become distributed as

$$\beta_k \sim N\left(\mathbf{0}, \sigma_{\beta}^2 diag\left(\frac{1}{1-\phi_k\lambda_i}\right)^{\theta_k}\right).$$

CHAPTER VII

APPLICATION OF METHODS TO ARCTIC RIVER RUNOFF DATA

Pan-Arctic and the River Runoff Data

Issues related to global climate change have continued to receive attention from both the scientific and political communities worldwide. Significant climate change would impact several global systems. One concern is its effect on global water supply. Although the Arctic Ocean contains only one percent of the global volume of seawater, it receives eleven percent of the world's river flow (Lammers, et. al., 2001). The pan-arctic hydrological system is believed to be particularly sensitive to global warming yet it is also hypothesized to be an important forcing of global climate. , Therefore it is of interest to monitor river flow in this region.

River discharge is monitored at several gaging stations in the pan-Arctic system. The observations taken at each of these sites between the years of 1960 and 1989 are the only ones considered here from the R-ArcticNET (see below) database. Interstation discharge values are computed by subtracting the discharge from all upstream gauges from the representative downstream discharge value. The area of region between river discharge gaging stations is the interstation area (figure 7.1). The station runoff is then computed as the interstation discharge divided by the interstation area.

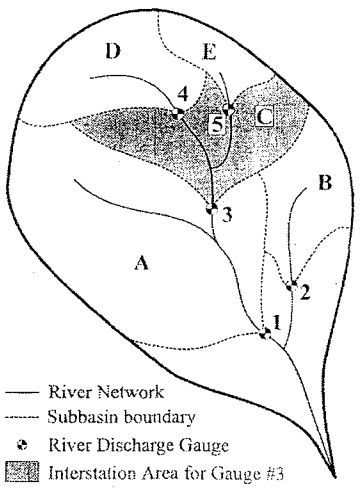

Figure 7.1: Pictorial view of an example drainage basin (Lammers et. al., 2001).

The region of interest, labeled the pan-Arctic, contains all land area draining into the Arctic Ocean, Hudson Bay, James Bay, Hudson Strait, and Bering Strait. This land area is primarily composed of Alaska and the northern regions of Canada, Russia, and other former Soviet states. This region can be further subdivided into individual river basins, depending on the application.

Data from this region has been compiled by the Water Systems Analysis Group in the Complex Systems Research Center at the Institute for Earth, Oceans, and Space at the University of New Hampshire. The compiled data set comes from several different sources and can be found on the web at

http://www.R-ArcticNET.sr.unh.edu.

The R-ArcticNET data bank contains measurements on river discharge and the computed river runoff values. Measurements were taken at stations distributed throughout the pan-Arctic region. Most gauges were between 48°N and 56°N with a sharp decline in the number of gauges north of 65°N.

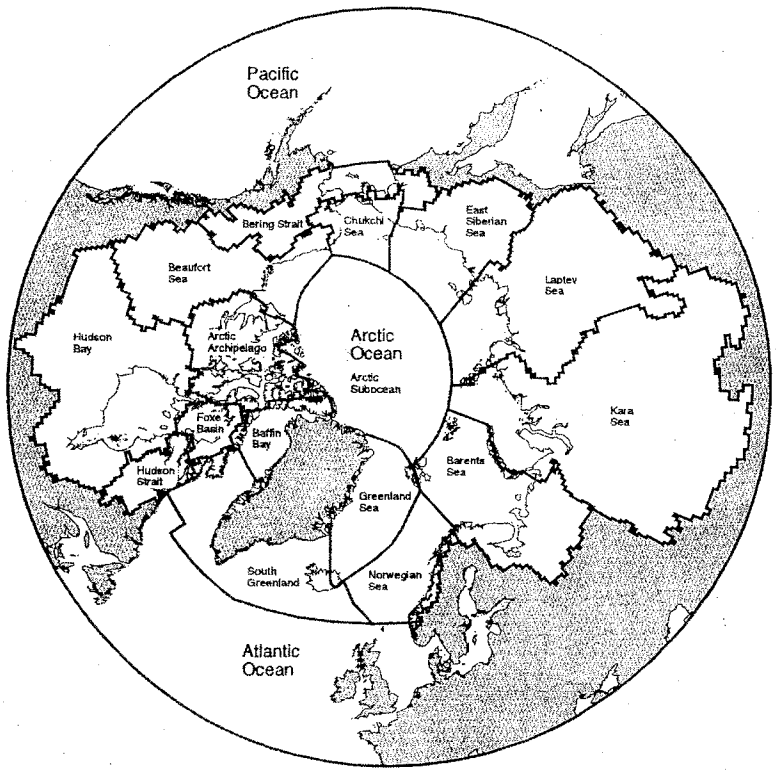

Figure 7.2: Map of the pan-Arctic region divided into sea basins (Lammers, et. al., 2001)

The purpose is to determine if river runoff is changing significantly and where these significant changes are occurring. So, it is of statistical interest to model the river runoff measurements over both space and time.

Previous Analyses of River Flow, River Discharge, and River Runoff

Several studies since the 1930's have looked at modeling some component of river flow over both space and time. Although the goals of these studies may not be all the same, they all address modeling the flow of water either into or out of a system over some period of time and at different locations.

One of these studies looks at statistical significance of the change in Amazon River discharge over the eighty-three year period between 1903 and 1985 (Richey, et. al., 1989). To analyze the temporal component, a time series analysis was performed using spectral methods to identify cycles in the discharge and stage levels at a single location, Manacapurú. The relationship between this data and data collected at a second location, Manaus, was established by performing a regressing on the two time series. Based on this regression, it was deemed that the data collected at Manacapurú was a good indication of overall Amazon discharge. The discharge was found to show no significant changes based on the slope of the linear regression of the deseasonalized time series against year with 95% confidence.

In some cases, properties of river flow and discharge are used as input variables to model other variables of interest. Variability in river discharge is analyzed by Horwitz (1978) as a comparison for fish species diversity trends. In particular, the homogeneity of river flow is tested by regressing annual mean and variability of flow against year. To determine the significance of within river variability (spatial variability), a linear regression of the variability at individual gaging stations on stream order and kilometers from headwater was performed. The results of these analyses were used as a ranking system for significance of variability in river flow.

Lammers et. al. (2001) look at the river runoff for the entire pan-Arctic. Using the data from R-ArcticNet, average annual runoff was computed for the thirty-year period from January 1960 to December 1989. These values were plotted over space, each point representing a gaging station. In addition, time series of average annual and seasonal runoff were averaged over stations for entire basins and were tested for significance using

the Mann-Kendall nonparametric test for trend. Since this analysis was performed on the same data and measurement of interest, it will be used as a basis of comparison for the following analyses.

Modeling River Runoff by Estimating Local Temporal Trend

A drawback of all three of the studies mentioned above is that they do not directly calculate any measure of spatial dependency, and thus do not take this into consideration when analyzing the temporal component of the data. The techniques of these studies involve identifying some sort of temporal trend, but limit themselves to entire basins instead of looking at individual locations so as to demonstrate local components of regional trends over space.

As suggested by Horwitz (1989), it is a natural extension to then look at temporal trends in river runoff at individual locations and view the significance of these trends over spatial location. This requires the estimation of simple linear regression coefficients over time at each gaging station. These temporal trends are then plotted over space and analyzed visually. Ordinary tests of significance for the slope parameter can be performed to determine if there is a significant increase or decrease in river runoff over the time observed.

A simple linear regression was performed on the runoff data in R-ArcticNet at each of the gaging stations with observations from 1960 to 1990. The resulting estimate of the slope parameter is then tested for significance at a 95% confidence level. All computations were done using the script LocalSLR.ssc written for S-PLUS version 6.2 which is included in Appendix B. The results of this analysis for the entire pan-Arctic

are shown in figure 7.3 and are shown for the Beaufort basin in figure 7.4, since this will be used for comparison to the other methods. Notice that for the Beaufort basin, there are distinct areas that demonstrate significant increase and others that show significant decrease in the river runoff over the approximately 30-year period. . Notice that this method indicates a relatively smooth spatial relationship.

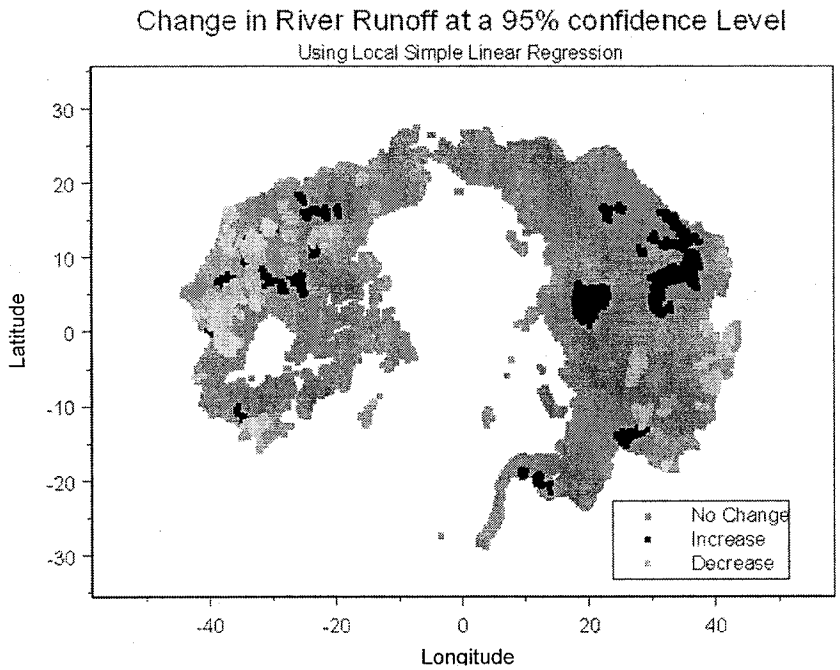

Figure 7.3: Significance of linear trends plotted over space for the entire pan-Arctic region.

There are several drawbacks to this technique. First, no measure of spatial dependence is estimated. That is, no assumption of spatial correlation is made. Each location or region is treated as completely independent of all others. Secondly, determining if there are trends over space is subjective because they are analyzed only visually. Lastly and most importantly, there is low statistical power in determining the significance of temporal trends estimated in any of these ways because they are based on a small number of data points (corresponding to the number of years over which the

observations were recorded). Therefore, analyses that pool all the data and model spatial dependencies explicitly will have increased power to detect a significant trend at each location - a more definitive result.

Modeling River Runoff using the Space-Time Hierarchical Model

To demonstrate the use of the space-time hierarchical model, the technique is applied to a single river basin, the Beaufort Sea basin. Recall from Chapter 6 and table 6.1, that even with only linear temporal trend assumed, there are several parameters to estimate. These are all necessary to model the behavior of the data over both space and time, but only beta is of interest to determine if river runoff is significantly decreasing or increasing.

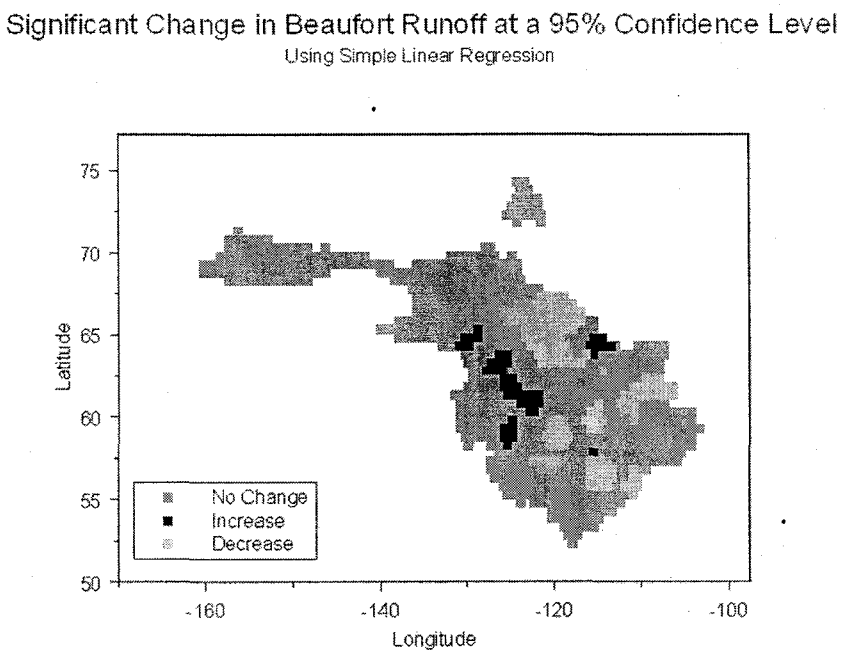

Figure 7.4: Significant increases and decreases in Beaufort river runoff at individual gauging stations over time using local simple linear regression.

In this case, embedding is not used and a lattice structure is not imposed on the Beaufort Sea basin. One approach to address the irregular boundary of the Beaufort grid is to use distance based weights. The maximum distance is set at 56. To determine this value, several numbers were tested and when plotted, 56 was the smallest value that resulted in all first order neighbors. In figure 7.5, the neighbor structure is shown with each vertex representing a data point and the lines indicating that connected points are neighbors.

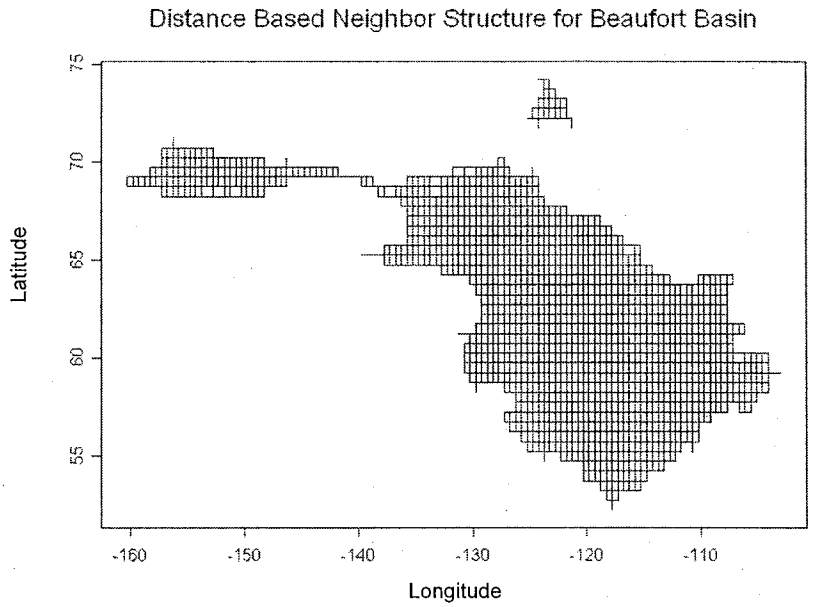

Figure 7.5: Beaufort distance-based neighbor structure indicating observation locations at vertices and line segments indicate two locations are neighbors.

The spatial portion of the analysis assumes the Pettitt, et. al. parameterization for the CAR model as a prior for the temporal regression parameter, β. For this case, the neighbor structure is setup using the script SpTimeCARdist.ssc and the parameters are all estimated as in the script SpTimeCARembed.ssc, both written for S-PLUS version

6.2. These codes are included in part in Appendix B. The output for the Beaufort Sea basin is shown in figure 7.6.

Notice that the output for the significance of temporal trend is not smooth like in the simple linear regression. This indicates that the distance-based approach of finding neighbors is not adequate to find the trends or that the CAR model does not adequately represent the smoothness of the runoff trends over space.

As a comparison, the analysis was done again using a S-PLUS generated neighbor structure. In this case, an expanded rectangular grid was overlaid on the Beaufort basin

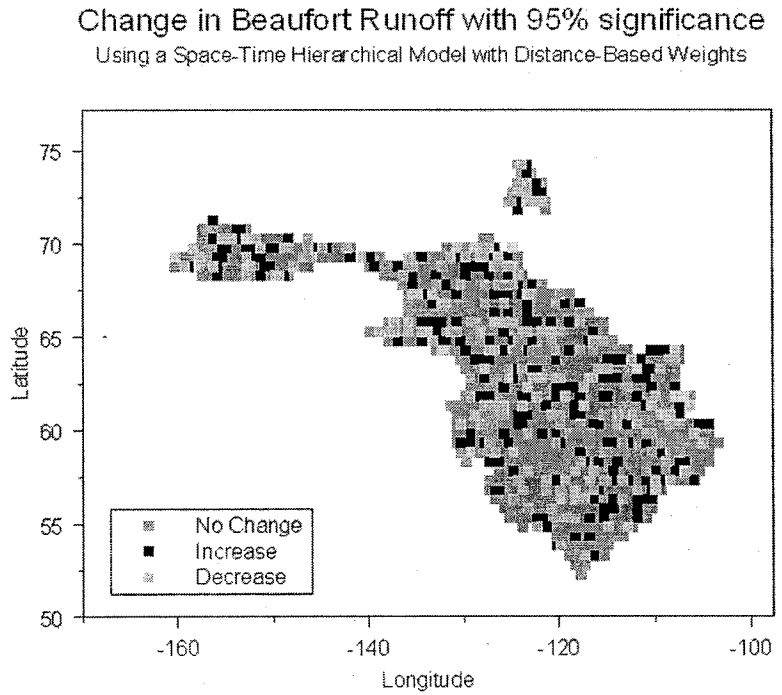

Figure 7.6: Significant increases and decreases in river runoff at individual gauging stations over time. Temporal Trend is assumed to have a CAR prior. Distance-based neighbor structure is used

for numbering purposes. First order neighbors were found and the uniform neighbor-weight matrix was generated using the neighbor.grid and spatial.weights functions built

into the S-PLUS spatial module. Once the neighbor-weight matrix was created, only the rows and columns corresponding to the Beaufort basin were used. The results are shown below in figure 7.7.

In both cases, using distance-based weights and the S-PLUS generated weights, the results are very similar. There is no discernable pattern in the locations that demonstrate significant increase or decrease. The only conclusion that can be drawn is that using the CAR priors for the temporal trend parameter does not adequately reflect the smoothness in the underlying process.

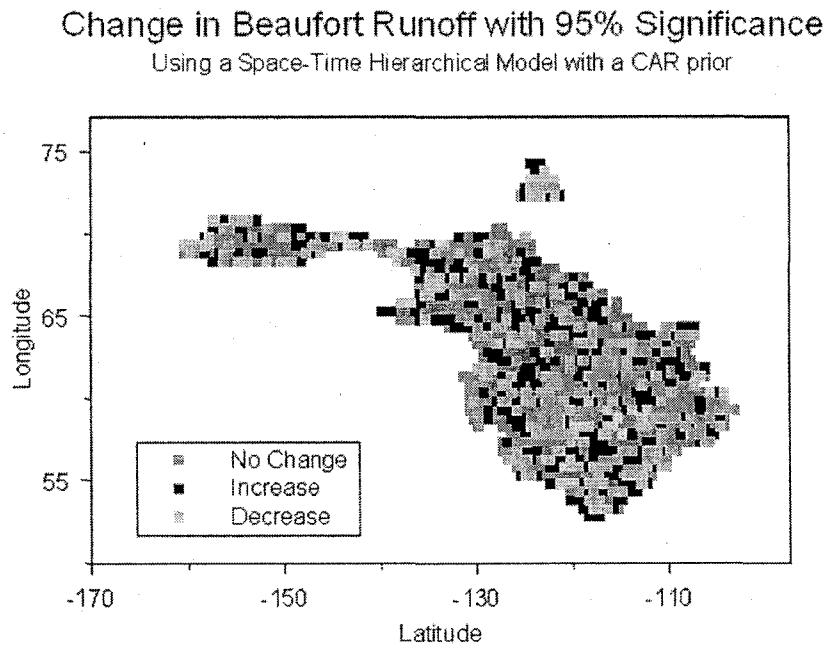

Figure 7.7: Significant increases and decreases in river runoff at individual gauging stations over time. Temporal Trend is assumed to have a CAR prior. S-PLUS generated neighbor structure is used.

The Addition of Circulant Embedding to the Space-time Model

To embed the Beaufort Sea basin within a regular grid, the spatial locations first need to be transformed to a lattice. Since the locations are equally spaced, this is easy to do.

The minimum and maximum longitudes and latitudes are used to created the rectangle so that it fits snugly around the observed locations and thus does not bog down computational time unnecessarily. The embedding is shown in figure 7.8. Here, all the additional points are assumed to have zero runoff. It is necessary to insert a number for each of these locations because the data cannot be transformed using the structure removing transformation if there are any missing values in the data set. This should not influence the analysis much since the runoff is significantly greater than zero at all spatial locations where measurements were taken. The only locations that may be affected are those on the perimeter of the Beaufort basin. It should be noted that the Beaufort basin was embedded in a grid with an odd number of rows and columns to guarantee that the total number of locations is odd so that use of fast Fourier transforms to obtain eigenvalues is valid.

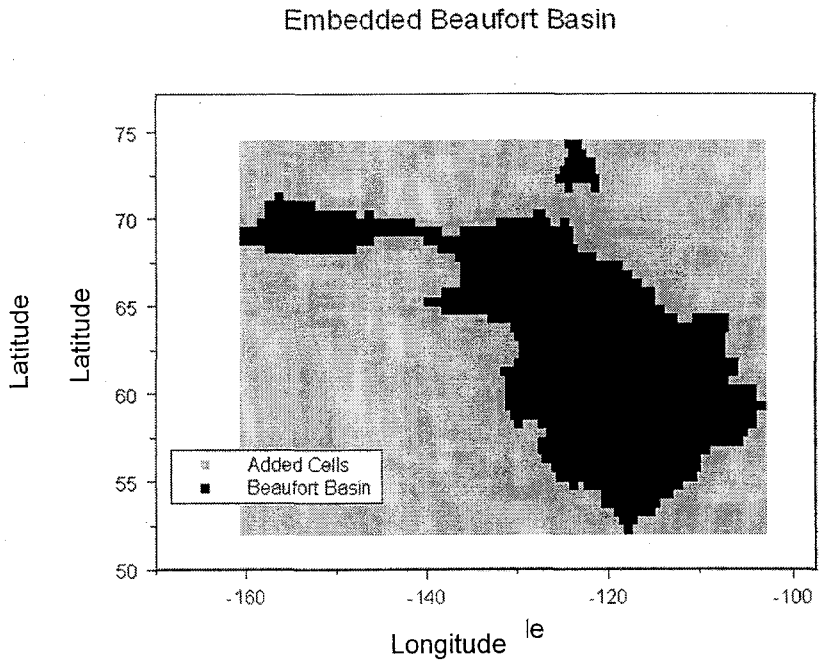

Figure 7.8: The Beaufort basin embedded in a rectangular grid.

Figure 7.9 below shows the results of the analysis performed assuming the CAR prior for the slope parameters in $\boldsymbol{\beta}$. One difference between the results of this analysis and the results from the simple linear regression is the number of locations that show significant trend. In the linear regression analysis, the majority of the Beaufort basin demonstrated no change in river runoff over time. Instead when the CAR prior is used with circulant embedding, there are many more locations with significant trend, and most of these show decrease in river runoff. It should be noted however that the pattern present in the simple linear regression analysis is similar to that using this model. The groupings of locations

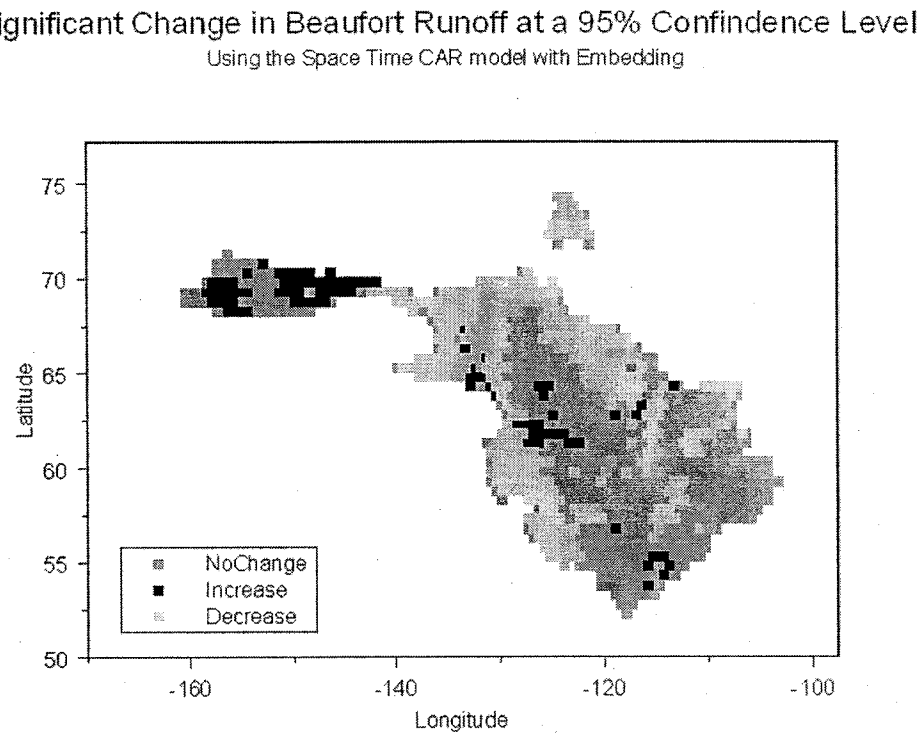

Figure 7.9: Significant increases and decreases in Beaufort river runoff at individual gauging stations over time using a space time hierarchical setup and circulant embedding with lattice-based first order neighbors. The Pettitt et. al. CAR parameterization is assumed as the prior for the linear temporal trend parameters.

with similar properties are merely expanded. Note that the tests of significance for the local simple linear regression are based on only 72 to 123 degrees of freedom (depending on the location) as compared to 641,699 degrees of freedom in the case of the space-time hierarchical model.

It should be noted that there are several locations along the perimeter of the Beaufort basin that show no significant change in runoff even though they are surrounded by locations that do indicate significant trend. This is due in part to the embedding and that the additional locations were given values of zero.

Modeling River Runoff using the EAR space-time model

In addition to embedding and incorporating the Pettitt et. al. parameterization of the CAR as the prior of the spatially distributed temporal trends, the parameter θ is included to allow for spatial smoothing as in the EAR model described in Chapter IV. The modifications made to the script SpTimeCARembed.ssc to estimate θ are included in Appendix B. The results of the analysis are shown in figure 7.10.

In this case, the decreasing trend extending down the middle of the Beaufort basin is an expansion of the regions indicating decreases in river runoff in the previous analyses. The area of increasing trend in the northwest-most portion of the basin and the region of decreasing trend along the western edge of the basin are slightly reduced. Since the distribution of river runoff trends is relatively smooth assuming the CAR prior, the addition of theta does not increase the smoothness much at all, but it does take into account much more of the information around it. Thus, this result is much more representative of the underlying process.

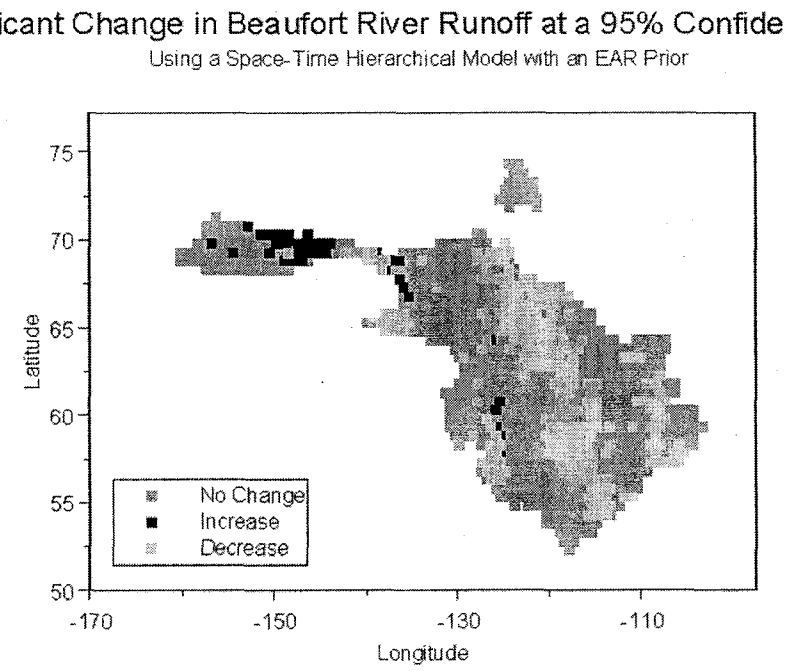

Figure 7.10: Significant increases and decreases in Beaufort river runoff at individual gauging stations over time using a space time hierarchical setup and circulant embedding with lattice-based first order neighbors. The EAR model is assumed as the prior for the linear temporal trend parameters.

CHAPTER VIII

CONCLUSIONS

In this thesis, methods of improving computational efficiency in estimating spatial parameters are proposed. In particular, the Pettitt, et. al parameterization for the CAR model is discussed. This parameterization results in a sparse symmetric neighbor weight matrix that is relatively easy to work with, but still uses a considerable amount of computation time when working with very large data like those in the environmental sciences. To compliment the computationally advantageous Pettitt, et. al. parameterization, a structure removing orthonormal transformation is described. This transformation is based on a singular value decomposition and results in the removal of spatial structure from the data.

Circulant embedding is also discussed as a method to decrease computation time for large data sets. Here, a smaller regular lattice structure with either a rectangular or irregularly shaped perimeter is embedded within a rectangular grid and wrapped around a torus. Numbering the locations sequentially so that they wrap around on each other allows for every location to have exactly the same number of neighbors. This results in a block circulant neighbor-weight matrix and thus greatly simplifies the calculation of eigenvalues and eigenvectors for the orthonormal transformation. The precision matrix is shown to be symmetric positive definite, making it a valid precision matrix.

The EAR model is proposed as an autoregressive model that accounts for smoothness of a process collected over space. This is just an extension of the Pettitt, et. al.

parameterization of the CAR model, but is shown to be equivalent to higher order CAR models when uniform weights are used. Thus, to model extremely smooth processes in space, use of the EAR model provides a more efficient analysis since it reduces the number of parameters needed in the model. Since the EAR model has as a special case the Pettit et. al. CAR model when theta is one, and for uniform weights is equivalent to higher order CAR models, it can be used in lieu of the CAR model and thus provides a better picture of the overall process.

These methods and strategies are then applied to a space-time hierarchical framework. The spatio-temporal model is described in general and a detailed formulation is given for a model with spatially varying temporal trends.

To demonstrate, several space-time analyses are performed on river runoff data compiled by the Water Systems Analysis Group at the University of New Hampshire. The observations in the data set are of river discharge, which can be used to calculate river runoff, over the entire Pan-Arctic region. Due to restrictions on computation time and available memory, the Beaufort basin alone is used in most cases.

As a point of comparison, and simple linear regression is performed over time at each individual location within the Beaufort basin and the slope is tested for significance. Locations are then categorized as showing significant increase, decrease, or no change in river runoff.

A drawback to not using embedding of any kind becomes immediately apparent when applying the space-time hierarchical model with a CAR prior. Given a region that is irregular in shape, it is extremely difficult to identify neighbors, regardless of order or numbering. In the case of the Beaufort basin, which has 1481 lattice points and is much

wider at top than it is on the bottom, it is extremely taxing to figure out the neighbors for each location by hand. In an attempt to perform the space-time analysis using the Pettitt, et. al. parameterization of the CAR as the prior for the spatially varying temporal trends, distance-based weights are used to identify neighbor structure.

If circulant embedding is used, the neighbors can be determined without using a distance-based approach. In this case, since a regular rectangular grid is formed, functions built into S-PLUS can be used to identify the neighbors, and then extra locations can be easily eliminated or disregarded.

Regardless of the method used, these analyses result in no apparent pattern in the locations that demonstrate significant trend. As noted in Chapter IV, the CAR model is the special case of the EAR model where $\theta = 1$ (see figure 4.1) and thus will not necessarily capture the smoothness of the underlying process.

When circulant embedding is added to the analysis, the time to compute the eigenvalues and eigenvectors of the neighbor weight matrix decrease significantly due to being able to use the fast Fourier transforms to find the eigenvalues. The results of this analysis are consistent with the patterns suggested by the simple linear regression, but appear to be more powerful as indicated by the increased number of locations that displayed significant change in runoff.

Finally, the EAR model is used as the prior for the spatially varying temporal trends. The results are similar to those when the CAR model with circulant embedding is used. Although there are some differences, the general regions that display significant trend in runoff are in approximately the same places, just smaller or larger in size.

For all the analyses performed, only a single basin was considered. It is desirable to be able to extend these models, especially the EAR model, to larger data sets. Since larger data sets require more computation time and more free memory, it may not be possible to perform these analyses using a software package such as S-PLUS. Instead, it may be possible to perform the analyses on pieces of a larger region, such as the basins in the larger pan-Arctic.

One approach being investigated is to use the data at locations in neighboring basins when performing analysis using circulant embedding. This would most certainly reduce if not eliminate the edges of the basin showing no significant increase as can be seen in figure 7.8. In the case of the pan-Arctic, only locations that are part of an ocean or other large body of water would have values of zero for runoff, whereas the other locations added due to circulant embedding would have their computed runoff values.

Doing this for each basin would involve two or more different estimates of linear trend at each location involved in circulant embedding. It is of interest to determine if the differences in these estimates are statistically significant. Then, it would need to be determined if the average of each estimate would provide a better picture of what is happening in each basin or if the estimate computed when looking at just that basin with embedding is sufficient.

LIST OF REFERENCES

Banerjee, S., Carlin, B.P., and Gelfand, A.E. (2004). *Hierarchical Modeling and Analysis for Spatial Data.* New York: Chapman & Hall/CRC Press.

Besag, J. (1974). "Spatial interaction and the statistical analysis of lattice systems (with discussion)." *Journal of the Royal Statistical Society B.* 36, 192-236.

Besag, Julian and Kooperberg, Charles (1995). "On conditional and intrinsic autoregressions," *Biometrika,* 82, 733-46.

Brockwell, P.J. and Davis, R.A. (2002). *Introduction to Time Series and Forecasting: Second Edition.* New York: Springer and Verlag.

Casella, George and George, Edward I. (1992). "Explaining the Gibbs Sampler," *The American Statistician*, 46, 3, 167-174.

Chatfield, Chris. (1996). *The Analysis of Time Series: An Introduction.* New York: Chapman and Hall.

Chib, S. and Greenberg, E. (1995). "Understanding the Metropolis-Hastings Algortithm," *The American Statistician*, 49, 4, 327-335.

Cliff, A.D. and Ord, J.K. (1981). *Spatial Processes: Models and Applications.* London: Pion Limited.

Cressie, N. (1991). *Statistics for Spatial Data.* New York: Wiley.

Gaudard, M., Karson, M., Linder, E. and Sinha, D. (1999). "Bayesian Spatial Prediction," *Environmental and Ecological Statistics*, 6, 147-182.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (2000). *Bayesian Data Analysis.*

Geman, S., and Geman, D. (1984). "Stochastic relaxation, Gibbs distribut6ions, and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995). "Adaptive Rejection Metropolis Sampling within Gibbs Sampling," *Applied Statistics*, 44, 4, 455-472.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.

Griffith, D. A., Layne, J. L., and Doyle, P.G. (1996). "Further Explorations of Relationships between Semi-Variogram and Spatial Autoregressive Models." General Technical Report RM-GTR-277, United States Department of Agriculture.

Handcock, M.S. and Wallis, J. R. (1994). "An Approach to Statistical Spatial-Temporal Modeling of Meteorological Fields." *Journal of the American Statistical Association,* 89,426,368-378.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods using Markov Chain and their Applications," *Biometrika*, 57, 97-109.

Hjort, N.L. and Omre, H. (1994). "Topics in Spatial Statistics," *Scandinavian Journal of Statistics*, 21, 289-357.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. New York: Cambridge University Press.

Horwitz, R.J. (1978). "Temporal Variability Patterns and the Distributional Patterns of Stream Fishes." *Ecological Monographs*. 48, 3. 307-321.

Kitanidis, P.K. and Lane, R.W. (1985). "Maximum Likelihood Parameter Estimation of Hydrologic Spatial Processes by the Gauss-Newton Method." *Journal of Hydrology,* 79, 53-71.

Lammers, R.B., Shiklomanov, A.I., Vörösmarty, C.J., Fekete, B.M., and Peterson, B.J. (2001). "Assessment of contemporary Arctic river runoff based on observational discharge records," *Journal of Geophysical Research*. 106, 3321-3334.

Linder, E. (2001). "Computer-Efficient Spatial Estimation and Interpolation Based on Conditional Gaussian Autoregressive Models," *Preprint.*

Mardia, K.V. and Marshall, R.J. (1984). "Maximum likelihood estimation of models for residual covariance in spatial regression," *Biometrika,* 71, 135-146.

Mardia, K.V. and Watkins, A.J. (1989). "On multimodality of the likelihood in the spatial linear model," *Biometrika*, 76, 289-295.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equations of State Calculations by Fast Computing Machines, " *Journal of Chemical Physics*, 21, 1087-1092.

Ord, K. (1975). "Estimation methods for models of spatial interaction," *Journal of the American Statistical Association*, 70, 120-126.

Pace, R.K. and Barry, R. (1996). "Sparse spatial autoregressions," *Statistics & Probability Letters*, 2158.

Pettitt, A.N., Weir, I.S., and Hart, A.G. (2002). "A Conditional Autoregressive Gaussian Process for Irregularly Spaced Multivariate Data with Application to Modeling Large Sets of Binary Data." *Statistics and Computing*. 12, 353-367.

Rice, J.A. (1995). *Mathematical Statistics and Data Analysis*. Belmont, California: Wadsworth Publishing Company.

Richey, J.E., Nobre, C., and Deser, C. (1989). "Amazon River Discharge and Climate Variability: 1903 to 1985." *Science*. 246, 4926. 101-103.

Rue, H. (2001). "Fast Sampling of Gaussian Markov Random Fields." *Journal of the Royal Statistical Society*. 63 part 2, 325-338.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. New York: Chapman & Hall / CRC.

Rue, H. and Tjelmeland, H. (2002). "Fitting Gaussian Markov random fields to Gaussian fields." *Scandinavian Journal of Statistics*. 29, p. 31 – 49.

Shiklomanov, A.I., Lammers, R.B., and Vörösmarty, C.J. (2002). "Widespread Decline in Hydrological Monitoring Threatens Pan-Arctic Research," *EOS (Transactions, American Geophysical Union)*, vol. 83,No. 2, pp. 13, 16, 17.

Stroud, J.R., Müller, P., and Sansó, B. (2001) "Dynamic models for spatiotemporal data," *Journal of the Royal Statistical Society*. 63 part 4, 673-689.

Vörösmarty, C.J., Federer, C.A., and Schloss, A.L., (1998). "Potential evoporation functions compared on US watersheds: Possible implications for global-scale water balance and terrestrial ecosystem modeling," *Journal of Hydrology*. 207, 147-169.

Vörösmarty, C.J., Fekete, B.M., Meybeck, M., and Lammers, R.B. (2000). "Geomorphometric attributes of the global system of rivers at 30-minute spatial resolution," *Journal of Hydrology*, 237, 17-39.

Wikle, C.K., Berliner, L.M., and Cressie, N. (1998). "Hierarchical Bayesian space-time models." *Environmental and Ecological Statistics*. 5, 117-154.

Wood, A. and Chan, G. (1994). "Simulation of Stationare Gaussian processes in [0, 1]d" *Journal of Compuational and Graphical Statistics*. 3 no. 4, 409-432.

APPENDICES

APPENDIX A

BAYESIAN CALCULATIONS FOR FULL CONDITIONAL POSTERIORS

All full conditional distributions are found using standard Bayesian calculations. These are summarized in any text on Bayesian methods such as Gelman, et. al. (1997). In the case of the full conditionals for the spatial hierarchical model, the transformed data, regression matrix, and process are used.

Full Conditional Distribution for the Transformed Spatial Process ($\tilde{\mathbf{z}}$)

In order to find the full conditional posterior distribution of the spatial process $\tilde{\mathbf{z}}$, it is first necessary to find the joint distribution,

$$p\left(\tilde{\mathbf{z}}, \tilde{\mathbf{y}} \middle| \sigma_y^2, \sigma_z^2, \boldsymbol{\beta}, \phi\right).$$

To do this, it is easiest to look at the distribution of the two vectors stacked. For simplicity, let

$$\eta_i = 1 - \phi\lambda_i$$

and

$$w_i = \frac{\sigma_z^2}{\sigma_z^2 + \sigma_y^2 \eta_i}.$$

Recall that

$$\tilde{\mathbf{y}} = \tilde{\mathbf{z}} + \boldsymbol{\varepsilon}$$

where

$$\boldsymbol{\varepsilon} \sim N\left(\mathbf{0}, \sigma_y^2 \mathbf{I}\right)$$

and

$$\tilde{\mathbf{z}} \sim N\left(\tilde{\mathbf{X}}\boldsymbol{\beta}, \sigma_z^2 diag\left(\frac{1}{\eta_i}\right)\right).$$

Note that the covariance of the data and the process is

$$Cov(\tilde{\mathbf{y}}, \tilde{\mathbf{z}}) = C(\tilde{\mathbf{z}} + \boldsymbol{\varepsilon}, \tilde{\mathbf{z}}) = Var(\tilde{\mathbf{z}}) + Cov(\boldsymbol{\varepsilon}, \tilde{\mathbf{z}}) = Var(\tilde{\mathbf{z}}).$$

Thus, the joint distribution can be written as

$$\begin{pmatrix} \tilde{\mathbf{y}} \\ \tilde{\mathbf{z}} \end{pmatrix} \sim N\left(\begin{pmatrix} \tilde{\mathbf{X}}\boldsymbol{\beta} \\ \tilde{\mathbf{X}}\boldsymbol{\beta} \end{pmatrix}, \sigma_z^2 \begin{pmatrix} \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) + \sigma_y^2 \mathbf{I} & \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) \\ \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) & \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) \end{pmatrix}\right).$$

For any joint distribution of the form

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right)$$

the conditional distribution can be expressed as

$$x_2 \left| x_1 \sim N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right).\right.$$

Thus, the full conditional distribution for the spatial process is normal with mean

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{z}|\mathbf{y}} &= \tilde{\mathbf{X}}\boldsymbol{\beta} + \sigma_z^2 diag\left(\frac{1}{\eta_i}\right)\left(\sigma_z^2 diag\left(\frac{1}{\eta_i}\right) + \sigma_y^2 \mathbf{I}\right)^{-1}\left(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}\right) \\ &= \tilde{\mathbf{X}}\boldsymbol{\beta} + diag(w_i)\left(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}\right) \end{aligned}$$

and variance covariance matrix

$$\begin{aligned} \sigma_{z|y}^2 &= \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) - \sigma_z^2 diag\left(\frac{1}{\eta_i}\right)\left(\sigma_z^2 diag\left(\frac{1}{\eta_i}\right) + \sigma_y^2 I\right)^{-1} \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) \\ &= \sigma_z^2 diag\left(\frac{1 - w_i}{\eta i_i}\right) \end{aligned}$$

Full Conditional Distribution for the Data Variance (σ_y^2)

By Bayes' rule, the full conditional distribution for the variance σ_y^2 can be written

$$p\left(\sigma_y^2 \middle| \tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_z^2, \phi\right) \propto p\left(\tilde{\mathbf{y}} \middle| \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_z^2, \phi, \sigma_y^2\right) p\left(\sigma_y^2\right).$$

Recall that $p\left(\sigma_y^2\right)$ is the inverse gamma prior and that

$$p\left(\tilde{\mathbf{y}} \middle| \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_z^2, \phi, \sigma_y^2\right) \sim N\left(\tilde{\mathbf{z}}, \sigma_y^2 \mathbf{I}\right).$$

This results in

$$p\left(\sigma_y^2 \middle| \tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_z^2, \phi\right) \propto \frac{1}{\left(\sigma_y^2\right)^{n/2}} e^{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{n}(y_i - z_i)^2} \cdot \left(\frac{1}{\sigma_y^2}\right)^{\alpha_y + 1} e^{-\frac{\gamma_y}{\omega_y^2}}$$

$$\propto \frac{1}{\left(\sigma_y^2\right)^{\frac{n}{2} + \alpha_y + 1}} e^{-\frac{1}{\sigma_y^2}\left(\gamma_y + \frac{1}{2}\sum_{i=1}^{n}(y_i - z_i)^2\right)}$$

which is inverse gamma with parameters

$$\alpha_{ypost} = \frac{n}{2} + \alpha_y$$

and

$$\gamma_{ypost} = \gamma_y + \frac{1}{2}\sum_{i=1}^{n}\left(y_i - z_i\right)^2.$$

Full Conditional Distribution for the Process Variance (σ_z^2)

The calculations for the process variance are similar to those done above for the data variance. In this case, though,

$$p\left(\sigma_z^2 \middle| \tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_y^2, \phi\right) \propto p\left(\tilde{\mathbf{y}}, \tilde{\mathbf{z}} \middle| \tilde{\boldsymbol{\beta}}, \sigma_z^2, \sigma_y^2, \phi\right) p\left(\sigma_z^2\right)$$
$$\propto p\left(\tilde{\mathbf{y}} \middle| \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_z^2, \sigma_y^2, \phi\right) p\left(\tilde{\mathbf{z}} \middle| \tilde{\boldsymbol{\beta}}, \sigma_z^2, \sigma_y^2, \phi\right) p\left(\sigma_z^2\right)$$

because σ_z^2 is one level farther into the hierarchical setup. Recall the joint distribution for $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$. Then the full conditional distribution for the process variance is the product of two normal distributions and an inverse gamma distribution, which simplifies to

$$p\left(\sigma_z^2 \middle| \tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\beta}}, \sigma_y^2, \phi\right) \propto \frac{1}{\left(\sigma_z^2\right)^{\frac{n}{2}+\alpha_z+1}} e^{-\frac{1}{\sigma_z^2}\left(\frac{1}{2}\sum_{i=1}^{n}\left(\eta_i\left(z_i - X\beta\right)^2\right)+\gamma_z\right)}.$$

Thus, the full conditional posterior distribution for the process variance is inverse gamma with parameters

$$\alpha_{zpost} = \frac{n}{2} + \alpha$$

and

$$\gamma_{zpost} = \frac{1}{2}\sum_{i=1}^{n}\left(\eta_i\left(z_i - X\beta\right)^2\right) + \gamma_z .$$

Full Conditional Distribution for the Trend Parameter ($\tilde{\boldsymbol{\beta}}$)

Calculations for finding the full conditional distribution for $\boldsymbol{\beta}$ are similar to those for the spatial process. Here, the joint distribution of $\tilde{\mathbf{z}}$ and $\tilde{\boldsymbol{\beta}}$ is

$$\begin{pmatrix} \tilde{\mathbf{z}} \\ \tilde{\boldsymbol{\beta}} \end{pmatrix} \sim N\left[\begin{pmatrix} \mathbf{X}\boldsymbol{\beta}_0 \\ \boldsymbol{\beta}_0 \end{pmatrix}, \begin{pmatrix} \sigma_z^2 diag\left(\frac{1}{\eta_i}\right) + \mathbf{X}\mathbf{T}_0\mathbf{X}^{\mathrm{T}} & \sigma_z^2\mathbf{T}_0\mathbf{X}^{\mathrm{T}} \\ \sigma_z^2\mathbf{T}_0\mathbf{X}^{\mathrm{T}} & \sigma_z^2\mathbf{T}_0 \end{pmatrix}\right]$$

where $\boldsymbol{\beta}_0$ and $\mathbf{T}_0$ are from the prior of $\boldsymbol{\beta}$. Then, the distribution of the process given all other parameters can be written

$$p\left(\tilde{\mathbf{z}}\middle|\tilde{\boldsymbol{\beta}},\sigma_z^2,\sigma_y^2,\phi\right)\sim N\left(X\boldsymbol{\beta}_0+\mathbf{T}_0\mathbf{X}^{\mathrm{T}}\mathbf{T}_0^{-1}\left(\tilde{\boldsymbol{\beta}}-\boldsymbol{\beta}_0\right),\sigma_z^2 diag\left(\frac{1}{\eta_i}\right)+\mathbf{X}\mathbf{T}_0\mathbf{X}^{\mathrm{T}}-\sigma_\beta^2\mathbf{T}_0\left(\mathbf{X}^{\mathrm{T}}\right)^2\right).$$

Recall that by Bayes' rule

$$p\left(\tilde{\boldsymbol{\beta}}\middle|\tilde{\mathbf{y}},\tilde{\mathbf{z}},\sigma_z^2,\sigma_y^2,\phi\right)\propto p\left(\tilde{\mathbf{y}}\middle|\tilde{\mathbf{z}},\sigma_z^2,\sigma_y^2,\phi\right)p\left(\tilde{\mathbf{z}}\middle|\tilde{\boldsymbol{\beta}},\sigma_z^2,\sigma_y^2,\phi\right)p\left(\tilde{\boldsymbol{\beta}}\middle|\sigma_\beta^2\right).$$

Thus,

$$p\left(\tilde{\boldsymbol{\beta}}\middle|\tilde{\mathbf{y}},\tilde{\mathbf{z}},\sigma_z^2,\sigma_y^2,\phi\right)\sim N\left(\mathbf{G}_{\mathrm{T}}\mathbf{H}^{-1}\hat{\boldsymbol{\beta}}+\left(\mathbf{I}-\mathbf{G}_{\mathrm{T}}\mathbf{H}^{-1}\right)\boldsymbol{\beta}_0,\sigma_\beta^2\left(\mathbf{I}-\mathbf{G}_{\mathrm{T}}\mathbf{H}^{-1}\right)\mathbf{T}_0\right)$$

where

$$\mathbf{H}^{-1}=\mathbf{X}^{\mathrm{T}}diag\left(\eta_i\right)\mathbf{X},$$

$$\mathbf{G}_{\mathrm{T}}=\left(\mathbf{H}^{-1}+\mathbf{T}_0^{-1}\right)^{-1},$$

and

$$\hat{\boldsymbol{\beta}}=\mathbf{H}\mathbf{X}^{\mathrm{T}}diag\left(\eta_i\right)\tilde{\mathbf{z}}.$$

For more information, see Hjort and More (1994).

Posterior Formulation for all Other Model Parameters

It is not possible to write the distribution for the spatial interaction parameter (ϕ), the temporal interaction from the space-time model (ρ), or the smoothness parameter from

the EAR model (θ) in closed form. Thus, it will be necessary to address this in the estimation scheme used. In each case, a Metropolis-Hastings step will be added.

APPENDIX B

SCRIPTS FOR SPLUS VERSION 6

All calculations used to analyze and demonstrate the techniques summarized were done using scripts written for S-PLUS version 6.2 unless otherwise noted. These are all included in this appendix. If the entire script does not appear, notes are included that explain how to extend the code to complete the task.

Script for Local Simple Linear Regression over Time

```
# Title:    LocalSLR.ssc
#
# Author:   Veronica Pocsik Hupper
# Updated:  June 18, 2003

# Description:
#
# Local SLR at each location over year and seasons -
   Outputs coordinates
#    slope terms (betas), the number of time points at each
   location,
#    the standard deviation of the slope estimates, lower
   and upper 95%
#    confidence bounds, and a logical indicator for
   significance of the
#    slope with 95% confidence (1 = significant).
#
# Each basin is treated individually for ease of
   computation only.  Below
#     is the programming for one basin only.  Change
   beaufort to any
#     other basin name to perform the same analysis for
   each basin.
```

```
#BEAUFORT BASIN (time and data both centered)

nloc <- nrow(beaufort.runoff)
ntime <- ncol(beaufort.runoff)
                                #Create a matrix of repeated
tt <- as.vector(1:ntime)        #time vectors, one column
                                #    for each location.
TT <- matrix(rep(tt,nloc),ncol=nloc)
TT[t(beaufort.runoff)=="NA"] <- NA

TT.min <- matrix(0,nrow=ntime, ncol=nloc)

for(k in 1:nloc) {
   TT.min[,k] <- min(TT[,k],na.rm=T)-1
}

#Center the time vector for each location

TT <- TT-TT.min
TT <- t(t(TT)-colMeans(TT,na.rm=T))

#Create a vector containing the
#    number of time points used
#    in each local regression.


Ind <- matrix(0,nrow=ntime,ncol=nloc)

Ind[t(beaufort.runoff)!="NA"] <- 1
beaufort.nreg <- colSums(Ind)
runoff.means <- rowMeans(beaufort.runoff, na.rm=T)
runoff.center <- beaufort.runoff-runoff.means

#Linear regression at each location

denom <- colSums(TT^2,na.rm=T)
beaufort.beta <- colSums(TT*t(runoff.center),na.rm=T)/denom
beaufort.MSE <- rowSums((runoff.center-
   beaufort.beta*t(TT))^2, na.rm=T)/(beaufort.nreg-1)
beaufort.sigbeta <- (beaufort.MSE^0.5)/(denom^0.5)

beaufort.err <- beaufort.sigbeta*qt(0.975,df=beaufort.nreg-
   1)
```

```
#Create a 95% confidence interval for beta at each
#  location.

beaufort.lowbeta <- beaufort.beta-(beaufort.err)
beaufort.upbeta <- beaufort.beta+(beaufort.err)

#Test for beta significantly different from 0 with 95%
#  confidence and determine if it is significantly
#  positive or negative.

betacrit <- rep(0, nloc)
betacrit <- qt(0.975,df=beaufort.nreg-1)*beaufort.sigbeta
beaufort.signbeta <- rep(".",nloc)
beaufort.signbeta[beaufort.beta > betacrit] <- "+"
beaufort.signbeta[beaufort.beta < (-1)*betacrit] <- "-"

# Output

beaufort.output <-
   cbind.data.frame(beaufort.cells$CellXCoord,beaufort.cells
   $CellYCoord,beaufort.beta,beaufort.sigbeta,beaufort.nreg,
   beaufort.lowbeta,beaufort.upbeta,beaufort.signbeta)
names(beaufort.output) <-
   c("Longitude","Latitude","Betas","SigBeta","nObs","Lower9
   5","Upper95","Signif95")


#Plot the sign of the slope over space

attach(beaufort.output)
plot(Longitude,Latitude,type='n')
text(Longitude,Latitude,beaufort.signbeta)
title("Change in Runoff for Arctic Archipelago Basin at a
   95% Confidence Level",xlab="Longitude",ylab="Latitude")
```

Script for Finding the Distance Based Neighbor Structure

```
# Title:     SpTimeCARdist.ssc
#
# Author:    Veronica Pocsik Hupper
# Updated:   June 15, 2005

# Description:
#
# Set up data and neighbor structure using distance
# based weights.  Then use same estimation and temporal
# setup as SpTimeCARembed.ssc below.


# Outputs:
#    Transformed data matrix to be used in estimating
#    the trend parameter.
#
# NOTE:  This is for the Beaufort Sea basin only.
#         Change beaufort to any other basin name to
#         do the same setup for each basin.



# Create function to calculate archangle and find distance
   based neighbor and weight matrix #

earthrad <- 6378.14

arcangle.findnb <- function(X,rmax)
{  lon <- X[,1]*pi/180
   lat <- X[,2]*pi/180
   arcangle <-
   acos(outer(cos(lat),cos(lat))*(outer(cos(lon),cos(lon))+o
   uter(sin(lon),sin(lon)))+outer(sin(lat),sin(lat)))
   arcangle <- arcangle[lower.tri(arcangle)]
   n <- nrow(X)
   rowind <- outer(1:n,rep(1,n))
   colind <- outer(rep(1,n),1:n)
   rowind <- rowind[lower.tri(rowind)]
   colind <- colind[lower.tri(colind)]
   findnb <-
   cbind.data.frame(rowind,colind,arcangle*earthrad)
   names(findnb)<-c('index1','index2','distances')
   findnb[findnb[,3] < rmax,]
   }
```

```
yrawmat <- as.matrix (beaufort.runoff)
ystd <- rowStdevs(yrawmat,na.rm=T)    # Center and
  standardize the data at each location #
ymean <- rowMeans(yrawmat,na.rm=T)
yrawmat <- (yrawmat-ymean)/ystd
n <- nrow(yrawmat)
XX <- beaufort.cells[,14:15]

### SETUP NEIGHBOR STRUCTURE ###

rmax <- 56
find.nb <- arcangle.findnb(XX,rmax)
gamuni <-
  spatial.neighbor(row.id=find.nb[,1],col.id=find.nb[,2],sy
  mmetric=T)
gamname <- "uniform"
gamma.nb <- switch(gamname, linear=gamlin,
  uniform=gamuni,reciprocal=gamrec)
plot(gamma.nb,xcoord=locs[,1],ycoord=locs[,2])
title("Distance Based Neighbor Structure for Beaufort
  Basin")


gammamat <- matrix(0, n, n)              ## Weight Matrix
  Setup ##
index <- as.matrix(gamma.nb[,1:2])
gammamat[index] <- gamma.nb$weights        ## This is not
  working right, so skipped for now.
gammamat <- gammamat + t(gammamat)
D <- diag(rowSums(gammamat))
SVDpos <- eigen(gammamat-D,symmetric=T)
lambdas <- SVDpos$values
Gpos <- as.matrix(SVDpos$vectors)
tGpos <- t(Gpos)

yrawmat[yrawmat=="NA"]<-0        # REPLACE NAs WITH ZEROS -
  don't need this when using interpolated data
ymat <- tGpos %*% yrawmat        # NAs in original data are
  causing this to be empty
y <- as.vector(ymat)             #  but with locations with
                                 #  all missing data
                                 #  removed, this
                                 #  seems okay.
```

Script for Finding the Distance Based Neighbor Structure

```
# Title:     SpTimeCARnoEmbed.ssc
#
# Author:    Veronica Pocsik Hupper
# Updated:   August 30, 2005

# Description:
#
# Set up data and neighbor structure using S-PLUS
# functions neighbor.grid and spatial.weights.  Then use
# same estimation and temporal setup as SpTimeCARembed.ssc
# below.


# Outputs:
#    Transformed data matrix to be used in estimating
#    the trend parameter.
#
# NOTE:  This is for the Beaufort Sea basin only.
#         Change beaufort to any other basin name to
#         do the same setup for each basin.


### Circulant Embedding - only for numbering purposes ###

locs <- beaufort.coord

maxX <- max(beaufort.coord[,1])
minX <- min(beaufort.coord[,1])
maxY <- max(beaufort.coord[,2])
minY <- min(beaufort.coord[,2])

nrowgrid <- (maxY - minY)/0.5 + 1
ncolgrid <- (maxX - minX)/0.5 + 1

xx <- seq(minX,  maxX  ,by = .5)
yy <- seq(minY, maxY ,by = .5)
grid <- expand.grid(xx,yy)
names(grid) <- c('CellXCoord','CellYCoord')

# Prepare Data to merge #

beaufort.ind <- rep(1,nrow(beaufort.coord))
beaufort.dat <- cbind(beaufort.ind,beaufort.coord)
names(beaufort.dat) <- c('Ind','CellXCoord','CellYCoord')
beaufort.dat <- cbind(beaufort.dat,beaufort.runoff)
```

```
# Merge

dat <- merge(grid,beaufort.dat,all.x=T)
dat.sort <-
   sort.col(dat,"@ALL",c("CellYCoord","CellXCoord"),ascendin
   g=T)
loc.nums <- seq(1, 5175, by = 1)
dat.sort <- cbind(loc.nums,dat.sort)

### SET UP SPATIAL WEIGHTS, EVALS, AND EVECS ###

beauf.nbrgrid.embed <-
   neighbor.grid(nrowgrid,ncolgrid,neighbor.type="third.orde
   r",
    weight.fun=NULL, matrix.fun=NULL, max.horiz.dist=5,
   max.vert.dist=5, use.pattern=F)


rowind <- which(dat.sort$Ind == 1)
smnb <- beauf.nbrgrid.embed
smnb <- smnb[smnb$row.id %in% rowind ,]
smnb <- smnb[smnb$col.id %in% rowind ,]

beauf.nbrmat <- spatial.weights(smnb, parameters=NULL,
   region.id=NULL)
beauf.nbrmat <- cbind(loc.nums,beauf.nbrmat)
beauf.nbrmat <- beauf.nbrmat[rowSums(beauf.nbrmat[,-1])>0 ,
   colSums(beauf.nbrmat)>0]

dat.sort.sm <-
   dat.sort[is.element(dat.sort[,1],beauf.nbrmat[,1]),]

gammamat <- beauf.nbrmat[,-1]

# Eigenvalues & Eigenvectors w/ SVD #

D <- diag(rowSums(gammamat))
SVDpos <- eigen(gammamat-D,symmetric=T)
lambdas <- SVDpos$values
Gpos <- as.matrix(SVDpos$vectors)
tGpos <- t(Gpos)
```

```
# Runoff Data #

yrawmat <- as.matrix (dat.sort.sm[,5:128])
ystd <- rowStdevs(yrawmat,na.rm=T)    # Center and
   standardize the data at each location #
ymean <- rowMeans(yrawmat,na.rm=T)
yrawmat <- (yrawmat-ymean)/ystd
n <- nrow(yrawmat)
XX <- dat.sort.sm[,2:3]


# Transform Data after standardizing #

yrawmat[yrawmat=="NA"]<-0          # REPLACE NAs WITH ZEROS -
   don't need this when using interpolated data
ymat <- tGpos %*% yrawmat          # NAs in original data are
   causing this to be empty
y <- as.vector(ymat)               #  but with locations with
   all missing data removed, this
                              #  seems okay.

t.ymat <- t(ymat)
```

Script for the Space-Time Hierarchical CAR Model with Embedding Annotated for Including the EAR Model

```
# Title:     SpTimeCARembed.ssc
#
# Author:    Veronica Pocsik Hupper
# Updated:   August 21, 2005

# Description:
#
# Temporal trend parameter (beta) is estimated using the
# space-time hierarchical model where the CAR (EAR) model
# is assumed as the prior for spatial distribution of the
# betas.  Circulant embedding is used

# Outputs:

#    slope terms (betas), the number of time points at each
#      location,
#    the standard deviation of the slope estimates, lower
#      and upper 95%
#    confidence bounds, and a logical indicator for
#      significance of the
#    slope with 95% confidence (1 = significant).
#
# NOTE:  Each basin is treated individually for ease of
#         computation only.  Below is the programming for
#         one basin only.  Change beaufort to any
#         other basin name to perform the same analysis for
#         each basin.


# Set up the time vector

tt <- 1:124    # time points
tt <- tt - mean(tt)
nt <- length(tt)

# Temporal Correlations

QQar <- diag(rep(0,nt))
QQar[abs(row(QQar)-col(QQar))==1]<- -1
lams.time <-eigen(QQar)$values
```

```
### Circulant Embedding ###

locs <- beaufort.coord

maxX <- max(beaufort.coord[,1])
minX <- min(beaufort.coord[,1])
maxY <- max(beaufort.coord[,2])
minY <- min(beaufort.coord[,2])

xx <- seq(minX,  maxX  ,by = .5)
yy <- seq(minY, maxY ,by = .5)
grid <- expand.grid(xx,yy)
names(grid) <- c('CellXCoord','CellYCoord')

# Prepare Data to merge #

beaufort.ind <- rep(1,nrow(beaufort.coord))
beaufort.dat <- cbind(beaufort.ind,beaufort.coord)
names(beaufort.dat) <- c('Ind','CellXCoord','CellYCoord')
beaufort.dat <- cbind(beaufort.dat,beaufort.runoff)

# Merge

dat <- merge(grid,beaufort.dat,all.x=T)

### SET UP SPATIAL WEIGHTS, EVALS, AND EVECS ###

# First row of first order neighbor indicator matrix

row1 <- rep(0,nrow(dat))
row1[2] <- 1
step <- 0.5
sx <- as.integer((grid[,1]-(minX-step))/step)
sy <- as.integer((grid[,2]-(minY-step))/step)
xnum <- max(sx)
row1[xnum+1] <- 1
n.embed <- nrow(grid)
row1[n.embed] <- 1
row1[n.embed-xnum+1] <- 1


# Eigenvalues #

eval <- Re(fft(row1)) - 4
```

```
# Eigenvectors #

n <- n.embed              # divide unit circle into
nhalf <- floor(n/2)          # n equal angles - look at
h <- 0:(nhalf)            # half at a time
omegas <- 2*pi*h/n


cjs <- t((2/n)^.5 *cos(outer(omegas[-1],0:(n-1))))
sjs <- t((2/n)^.5*sin(outer(omegas[-1],0:(n-1))))
c0 <- rep(1,n)/(n^.5)
Evecs <- cbind(c0,matrix(nrow=n,ncol=(n-1)))
Evecs[,seq(2,(n-1),by=2)] <- cjs
Evecs[,seq(3,n,by=2)] <- sjs

# Transform Data after standardizing #

ymat <- dat[,4:ncol(dat)]
ystd <- rowStdevs(ymat,na.rm=T)
ymean <- rowMeans(ymat,na.rm=T)
ymatstd <- (ymat-ymean)/ystd


# REPLACE NAs WITH ZEROS - don't need this when using
#   interpolated data.

ymatstd[ymatstd=="NA"]<-0

# Transformation

ymat <- t(Evecs) %*% ymatstd
y <- as.vector(ymat)
t.ymat <- t(ymat)

### ESTIMATION PROCEDURE ###

lambdas <- eval

# Priors

beta0 <- rep(0,n)                    #slope prior#
alpha0 <- 0.1                        #prior for the variance
gam0 <- 1.1                          #
alpha0.phi <- 0.1                    #spatial parameter priors
gam0.phi <- 1.1                      #
alpha0.phibeta <- 0.1
gam0.phibeta <- 1.1
```

```
# mu.u <- 0                    #u=log(theta) is EAR parameter
# prsig.u <- 0.1

alphapost <- alpha0 + n*nt/2 + n/2 #post for the variance -
   depends on sample size#

# The Chain
# Set up the MCMC parameters
##############################

n.burn <- 300
n.chain <- 1000
ntotal <- n.burn + n.chain            ##Initializing
   paramters
sig2 <- vector(length=ntotal)
sig2beta <- vector(length=ntotal)
#phi <- vector(length=ntotal)
   phi <- rep(0,ntotal)
phibeta <- vector(length=ntotal)
rho <- vector(length=ntotal)
# theta <- vector(length=ntotal)      # Only use for EAR
# u <- vector(length=ntotal)          #   Model


# Tuning parameter for Metropolis
sig.phi  <- 0.2
rangeuni <- 0.2
sig.u <- 0.2

# Initiating the Chain

sig2[1] <- 1
sig2beta[1] <- .1             #starting point for the chain
     phi[1] <- 0
rho[1] <- .025
phibeta[1] <- .05
theta[1] <- 4
u[1] <- log(theta[1])
Eofbeta <- rep(0,n)
varofbeta <- rep(0,n)
#count.phi <- 0
count.phibeta <- 0
count.rho <-0
count.theta <- 0          # How many are accepted
```

```
#  MCMC  MCMC   MCMC  MCMC
##############################

# Updating
for (i in 1:(ntotal-1))
{  # i <- 1
   print(i)

# Preliminary Calculations
   etas <- 1-phi[i]*lambdas
   xis <- 1 - phibeta[i]*lambdas

   #If Using EAR model
   # xis <-(1 - phibeta[i]*lambdas)^theta[i]

   deltas <- 1 - rho[i]*lams.time

# Gibbs Step for beta and Calculations of Eofbeta and
   varofbeta

   Qar <- diag(nt) + rho[i]*QQar
   cstar <- as.vector(t(tt) %*% Qar %*% tt)
   betahat <- colSums(as.vector(Qar %*% tt) * t.ymat)/cstar
   denom <- cstar*etas + xis/(sig2beta[i]/sig2[i])
   varbeta <- sig2[i]/denom
   meanbeta <- (cstar*etas/denom)*betahat +
   (xis/(sig2beta[i]/sig2[i]))*beta0
   beta <- varbeta^.5 * rnorm(n) + meanbeta

# Recursive updating of mean and var of beta
   if (i > n.burn)
   {
      Eofbeta <- Eofbeta+meanbeta/n.chain   #varofbeta=2nd
   moment
      varofbeta <- varofbeta + varbeta/n.chain +
   (meanbeta^2)/n.chain
   }

# Residuals and other intermediate calculations
   resid <- as.vector(y-kronecker(tt,beta))
   lagvec <- c(rep(0,n),resid[1:((nt-1)*n)]) +
   c(resid[(n+1):(nt*n)],rep(0,n))
   SS <- sum(resid^2 * rep(etas,nt) -
   rho[i]*resid*rep(etas,nt)*lagvec)

# 6 Metropolis Step for phibeta
    phibetanew <- rnorm(1,mean=phibeta[i],sd=sig.phi)
```

```
   if (phibetanew > 0)
   {
   xis.new <- 1-phibetanew*lambdas
   betadiff2 <- (beta-beta0)^2
   phibetadet <- sum(log(xis))
   phibetanewdet <- sum(log(xis.new))
   AR <- exp(0.5*((phibetanewdet-phibetadet)+
   sum(betadiff2*(xis-xis.new))/(sig2beta[i]))
    + (alpha0.phibeta-1)*(log(phibetanew)-log(phibeta[i]))
   +gam0.phibeta*(phibeta[i] - phibetanew))
   U <- runif(1)
   if (U <= AR)
   { phibeta[i+1] <- phibetanew
     xis <- xis.new
     count.phibeta <- count.phibeta + 1
   }
     else phibeta[i+1] <- phibeta[i]
   }
   else phibeta[i+1] <- phibeta[i]

# Metropolis Step for Theta (via u)

   u.new <- rnorm(1,mean=u[i], sd=sig.u)
   theta.new <- exp(u.new)
   xis.new <- (1-phibeta[i+1]*lambdas)^theta.new
   thetadet <-sum(log(xis))
   thetanewdet <- sum(log(xis.new))

   # ACCEPTANCE RATIO #

   thetadet.diff <- thetanewdet-thetadet
   sum.betaxis <- sum(betadiff2%*%(xis-xis.new))
   lik1 <- ((u[i]-mu.u)^2)/sig.u^2
   lik2 <- ((u.new-mu.u)^2)/sig.u^2
   AR <- exp(0.5*(thetadet.diff+sum.betaxis)/sig2beta[i])
        +exp(0.5*(lik1-lik2))

   U <- runif(1)
   if (U <= AR)
   {
     u[i+1] <- u.new
     theta[i+1] <- theta.new
     xis <- xis.new
     count.theta <- count.theta+1
   }
   else
```

```
      u[i+1] <- u[i]

   theta[i+1] <- exp(u[i+1])

# Gibbs Step for sig2beta

   SSbeta <- sum(xis*((beta-beta0)^2))
   sig2beta[i+1] <- 1/(rgamma(1,(n/2-1),SSbeta/2))


# 7 Metropolis Step for rho

   cc <- min(rangeuni,0.5*abs(1-abs(rho[i])))
   rhonew <- rho[i]+ .5*runif(1,-cc,cc)
   rhodet <- sum(log(deltas))
   deltas.new <- 1 - rhonew*lams.time
   rhonewdet <-  sum(log(deltas.new))
   diffSS <- sum(- rho[i]*resid*rep(etas,nt)*lagvec  +
   rhonew*resid*rep(etas,nt)*lagvec)

   AR <- exp(0.5*(diffSS/sig2[i] + n*(rhonewdet-rhodet)) )

   U <- runif(1)
   if (U <= AR)
   {  rho[i+1] <- rhonew
      deltas <- deltas.new
      count.rho <- count.rho + 1
   }
      else rho[i+1] <- rho[i]

# Gibbs Step for sig2

   gampost <- gam0 + .5 * SS + SSbeta/(2*sig2beta[i+1])
   sig2[i+1] <- 1/rgamma(1,alphapost,gampost)

}

# Reviewing the results     .

count.phibeta; count.rho ; count.theta

# Transform back

varofbeta <- varofbeta-Eofbeta^2

###  FINAL TRANSFORMED ESTIMATES ###
```

```
dvb <- diag(varofbeta)
Edvb <- Evecs %*% dvb
EdvbE <- Edvb %*% Evecs
Varofbeta <- diag(EdvbE)
Meanofbeta <- t(Evecs) %*% Eofbeta


### TEST BETAS FOR SIGNIFICANCE AND PLOT BY
###  LOCATION TO COMPARE TO BETAS ###
###  GENERATED USING OLS REGRESSION


beaufort.meanbeta <- as.vector(Meanofbeta)

#  Is estimated beta different from 0 with 95% confidence?

betacrit <- qnorm(0.975,mean=0,sd=1)*(Varofbeta^0.5)


#  Determine if it is significantly positive or negative.

beaufort.signbeta <- rep(".",nrow(grid))
beaufort.signbeta[beaufort.meanbeta > betacrit] <- "+"
beaufort.signbeta[beaufort.meanbeta < (-1)*betacrit] <- "-"

beaufort.output <-
   cbind(dat[,1:3],beaufort.meanbeta,beaufort.signbeta)
names(beaufort.output) <-
   cbind("CellXCoord","CellYCoord","ind","beta","signbeta")

menuSubset(data = beaufort.output, subset.expression =
   "ind==\"1\"", subset.columns = "<ALL>",
    result.type = "Data Set", subset.col.name = "Subset",
    save.name = "beaufort.only.output", show.p = F)
```